THE DERIVE - NEWSLETTER #19

THE BULLETIN OF THE

USER GROUP

Contents:

[1] **Elementary Linear Algebra with *DERIVE*,** R J Hill and Th A Keagy
Chartwell – Bratt, 1995, ISBN 0 86238 403 6
(This is a textbook with *DERIVE* fully integrated. *DERIVE* is used as a natural tool, 370 pages)

[2] **Technology in Mathematics Teaching,** L Burton and B Jaworsky, ed.
Addison – Wesley, 1995, ISBN 3 89319 857 1
(This is not a *DERIVE*-book. It addresses general issues raised by the introduction of technology into the teaching and learning of mathematics, 496 pages)

Two "classics" of *DERIVE* literature are now available in a German translation:

[3] **Mathematik entdecken mit *DERIVE* – von der Algebra zur Differentialrechnung,**
Jerry Glynn, Birkhäuser, 1995, ISBN 3 7643 5001 6, 154 Seiten

[4] **Mathematik lernen mit *DERIVE*,** J Berry, A Watkins, E Graham, Birkhäuser, 1995, ISBN 3 7643 5193 4, ca 360 Seiten

[5] **Introductie *DERIVE* 3.0,** Paul Drijvers, STAM Techniek, 1995, ISBN 90 401 0640 1
(A Dutch *DERIVE* book from Paul, a *DERIVIAN* from the first days.)

Exchange for DERIVE Teaching materials in the DNL *The wheel has not to be invented twice.*	**Börse für DERIVE Unterrichts-materialien im DNL** *Das Rad muss nicht zweimal erfunden werden.*

I can offer:

Binomial Theorem, GCD & LCM, System of Coordinates (in English and in German as well)

Second North American DERIVE User Group Meeting

I have the pleasure to announce the 2nd *DERIVE* User Group meeting in the United States of America. We will meet at the occasion of the

ICTCM 1995, November 16-19, 1995 in Houston, Texas.

Jerry Glynn from MathWare in Urbana, IL, has promised to help organizing this meeting. If you will not receive more details about the DUG meeting until the beginning of the conference you will find information at the booth of Math-Ware. I am very glad that - supported by Soft Warehouse Hawaii - I am able to be present at this event. And I hope to meet many of our American members and the European members, who will attend this important conference.

I am looking forward to meeting you in Texas.

Josef

Liebe DUG-Mitglieder,

Da ich diesen Newsletter rasch fertigstellen möchte, wird mein Letter kürzer als gewöhnlich ausfallen.

Der Sommer war ausgefüllt mit DERIVE- und anderen Reisen. Außerdem musste die deutsche Übersetzung des Buchs "Learning Mathematics through DERIVE" abgeschlossen werden.

Das herausragende Ereignis des Sommers war sicherlich die International DERIVE Conference in Honolulu. Viele der 35 Derivianer aus aller Welt nahmen die Konferenz zum Anlass, diesen wunderschönen Teil unserer Erde länger zu besuchen. Und ich denke, dass keiner seinen Aufenthalt auf der einen oder anderen Insel bereut hat. Meine Frau und ich besuchten sechs Inseln und hatten einen unvergesslichen Urlaub. An dieser Stelle herzlichen Dank im Namen aller teilnehmenden DUG Mitglieder an Soft Warehouse Hawaii für die Vorbereitung und Unterstützung der Konferenz, sowie für die Gastfreundschaft und Betreuung während der Tage in Honolulu. Mahalo nui.

In diesem Herbst wird bestimmt wieder ein Meilenstein in der Entwicklung der Computer Algebra gesetzt. Mit dem TI-92 kommt ein "Taschenrechner" auf den Markt, der ein CAS, sehr ähnlich DERIVE, mit allen seinen Vorzügen implementiert hat. Aber nicht nur das, auch CABRI-Geometre, ein Texteditor und eine Tabellenkalkulation - die nicht nur numerisch. sondern auch symbolisch arbeiten kann -finden sich in diesem Zauberkasten, der in USA ca 200 $ kosten soll.

Ich habe DERIVE für WINDOWS gesehen!
Teresa, eine charmante und attraktive junge Dame, arbeitet emsig daran, DERIVE so wie wir es alle schätzen, in die WINDOWS Umgebung zu übertragen. Daneben ist sie eine äußerst begabte Hula-Tänzerin. Aloha Teresa, aus Österreich und herzliche Grüße an Deine kongenialen Tanzpartnerinnen.

Bärbel Barzel hat angekündigt, dass ein ausführlicher Tagungsband der DERIVE Days Düsseldorf herausgegeben wird. Sie werden sicher im nächsten DNL darüber informiert.

Herzliche Grüße

Dear DUG Members,

This letter will be shorter than usual because I want to finish the Newsletter in time.

Summer was spent with DERIVE- and other journeys. Moreover the German translation of "Learning Mathematics through DERIVE" had to be finalized.

*Certainly the most important event of this summer was the International DERIVE Conference in Honolulu. On this occasion many of the 35 Derivians from all over the world visited this wonderful part of our globe. And I think that nobody has regreted his stay on one or the other island. My wife and I were on six islands and it was an unforgetable holiday.
I want to express our special thanks to Soft Warehouse Hawaii for preparing and supporting the conference and for the hospitality and care during our stay in Honolulu on behalf of all DUG members present at the conference. Mahalo nui.*

In this fall will be set another milestone in the evolution of Computer Algebra. With the TI-92 we will have a pocket calculator with a regular DERIVE implemented. Additionally CABRI-Geometre, a text editor and a spread sheet - with numerical and symbolic abilities - can be used on this magic box. For the US a market price of about 200$ is announced.

I have seen DERIVE for WINDOWS!!!
Teresa, a really charming and attractive lady - is busy to transfer all that what we are appre- ciating in DERIVE into a WINDOWS environ- ment. Besides that she is a very gifted hula dancer. Aloha Teresa, from Austria and best greetings to your congenial hula partners.

Best regards
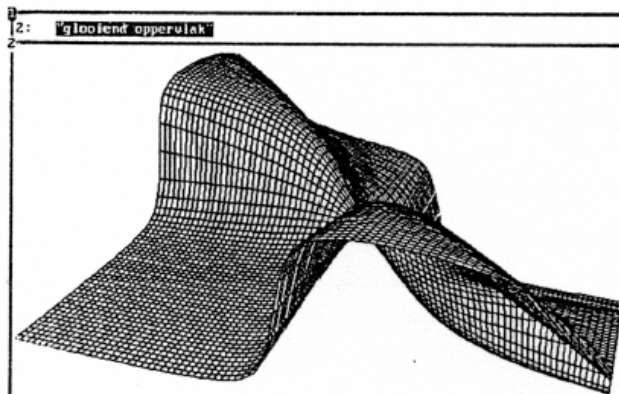
The *DERIVE-NEWSLETTER* is the Bulletin of the *DERIVE User Group*. It is published at least four times a year with a contents of 30 pages minimum. The goals of the *D-N-L* are to enable the exchange of experiences made with *DERIVE* as well as to create a group to discuss the possibilities of new methodical and didactical manners in teaching mathematics.

Editor: Mag. Josef Böhm
A-3042 Würmla
D´Lust 1
Austria
Phone: 43-(0)2275/8207

Contributions:
Please send all contributions to the Editor. Non-English speakers are encouraged to write their contributions in English to reinforce the international touch of the *D-N-L*. It must be said, though, that non-English articles are very welcome nonetheless. Your contributions will be edited but not assessed. By submitting articles the author gives his consent for reprinting it in *D-N-L*. The more contributions you will send, the more lively and richer in contents the *DERIVE Newsletter* will be.

Preview: (Contributions for the next issues):

Stability of systems of ODEs, Kozubik, SLO
Graphic Integration, Probability Theory, Linear Programming, Böhm, A
LOGO in DERIVE, Lechner, A
DREIECK.MTH, Wadsack, AUS
IMP Logo and Misguided Missiles, Sawada, HAWAII
3D Geometry, Reichel, AUS
Parallel- and Central Projection, Böhm, AUS
Müller´s Method to solve univariate equations a.o., Speck, NZL
Vector and Vector Indices Sorting, Biryukov, RUS
Algebra at A-Level, Goldstein, UK
Tilgung fremderregter Schwingungen, Klingen, GER
Utility for Complex Dynamic Systems, Lechner, A
Some Improvements on the Resolution of ODEs, Fuster, E
Notes on DERIVE 2.6 functions and limits, Speck, NZL
and

Setif, FRA; Vermeylen, Belgium; Lymer, FRA; Leinbach, USA, Aue, GER; Weth, GER, Wiesenbauer, AUT; Keunecke, GER
and messages from the derive-news@mailbase.ac.uk

DERIVE 3.x. Bugs or not

Preparing this issue I had three times the problem, that functions which did very well with versions 2.5x led to a desaster using *DERIVE* 3.x. (Spline-function and Bézier-functions). I found similar problems with some own functions, unfortunately once when I tried to demonstrate a difficult file during a presentation. During my stay in Honolulu I found some occasions to talk with Albert Rich about these "incompatibilities". One problem could be solved immediately. See the explanation in the Bézier contribution concerning the MB-function, page 31. This case will not occur very often. The other one has much more chances to happen. Back home from the holidays I found a letter from Albert Rich which I will cite:

Enclosed is the DERIVE XM version 3.06 diskette I promised to send you. It fixes the bug caused by the user-defined function that tried to extract the 0th element of a vector. For example, simplifying `0*element([x,y],0)` or `0*[x]SUB 0` used to hang DERIVE before a version 3.06.

Concerning the problem with SPLINE.MTH: Often a dummy variable needs to be introduced in a function's definition that is not a formal argument of the function. For example, the following function that sums the square of a vector's elements introduces the dummy variable i:

```
sumsqr(v) := sum((v SUB i)^2, i, 1, DIMENSION(v))
```

If one function calls another and both functions introduce the same variable, the introduced variable's value may not be what was intended. Therefore, introduced variables in functions that call each other should be given unique names.

The problem with spline.MTH was caused by using z_ as the introduced dummy variable in several function definitions. The revised SPLINE.MTH included on your diskette corrects the problems by using unique variable names for introduced variables.

Let me know as soon as you or a DUG member finds a new or old problem with DERIVE.

Many thanks, Albert. By the way I have to mention that Albert is very interested in the DNLs as one of his most important sources for improving DERIVE. So go on, hunting for problems and bugs, Albert is waiting.

Nevertheless, I cannot find it very userfriendly if functions loose their functionality by upgrading DERIVE. That might cause very surprised eyes during a demonstration of DERIVE's capabilities. Both changes mentioned above are surely justified by mathematical exactness. But I hope for us all that those changes will be at least announced in a README-file.

What you have to do now with the spline-file on page 18 is the following: Substitute the variable z_ in the functions SPLINE-D (e.g. by a_), SPLINE-B (by b_) and SPLINE_AUX (by c_). I could have done it for you. But I think that you have to see this possibility of making yourself a lot of troubles and how to resolve them. (The diskette of the year will contain the revised version). *Is obsolete with DERIVE 6.10, the files are ok, Josef*

Wolfgang Pröpper. Nürnberg, Germany

During the last weeks - testing the new TI-92 - I found an interesting question:

DERIVE seems to dislike the product $\sqrt{a} \cdot \sqrt{b}$. There is no trick to bring it into the form $\sqrt{a \cdot b}$. This would not be so bad, but if I consider the product $\sqrt{1-x^2} \cdot \sqrt{\dfrac{1+x}{1-x}}$, which is defined for -1 ≤ x < 1, and can be simplified by hand using common sense to (1+x), it becomes problematically. *DERIVE* refuses all attempts simplifying this expression. The plot is correct only for -1 ≤ x < 1. The plot shows $-|1+x|$.

The same will occur with expression $\sqrt{x^2-1} \cdot \sqrt{\dfrac{1+x}{x-1}}$.

DNL: Try to substitute $a = -2$, $b = -8$, then you will find that, following the rule drawing roots before multiplying expressions, there is no identity in your expressions: $\sqrt{-2} \cdot \sqrt{-8} = i\sqrt{2} \cdot i\sqrt{8} = -4$ but $\sqrt{(-2) \cdot (-4)} = 4$.

If you declare the variables nonnegative then you are able to proof the identity. The same idea can be applied to your other expression. I don't understand, why in this case *DERIVE* fails to show the identity for $x \in [-1, 1)$.

This was DERIVE 3.xx
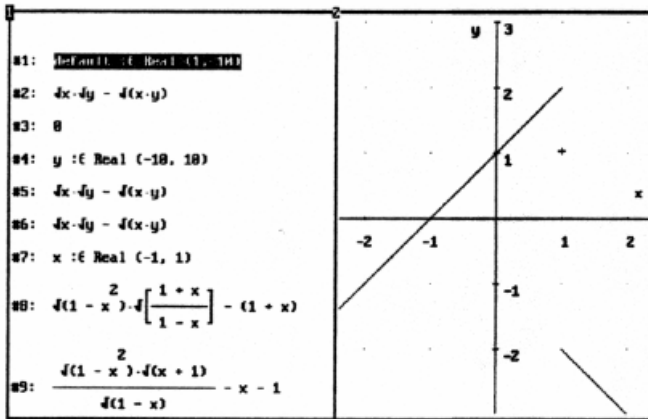
This is DERIVE 6.10:

#1: $\sqrt{x} \cdot \sqrt{y} = \sqrt{x} \cdot \sqrt{y}$

#2: $\sqrt{x} \cdot \sqrt{y} - \sqrt{(x \cdot y)} = \sqrt{x} \cdot \sqrt{y} - \sqrt{(x \cdot y)}$

#3: $x \in \text{Real } (0, \infty)$

#4: $\sqrt{x} \cdot \sqrt{y} = \sqrt{x} \cdot \sqrt{y}$

    but:

#5: $\sqrt{x} \cdot \sqrt{y} - \sqrt{(x \cdot y)} = 0$
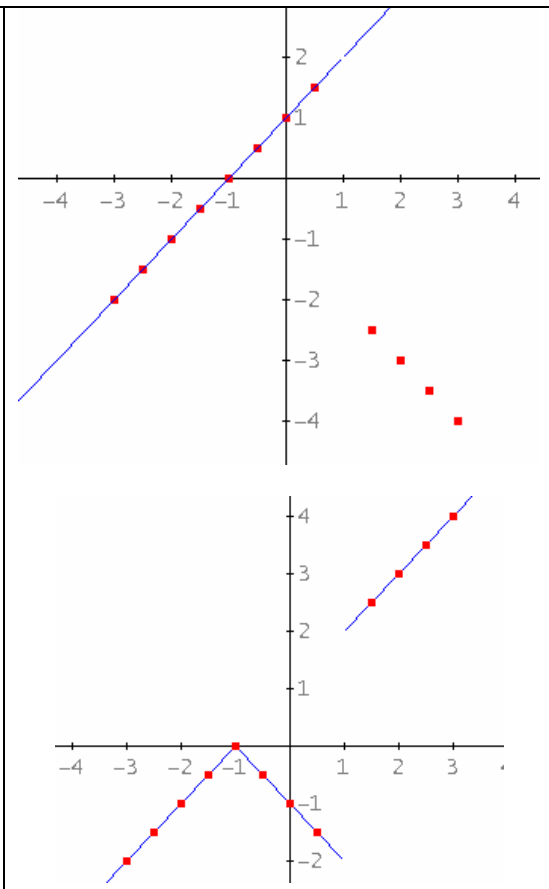

This is DERIVE 6.10:

#6: $x \in \text{Real}$

#7: $\sqrt{(1 - x^2)} \cdot \sqrt{\left(\dfrac{1 + x}{1 - x} \right)} = \sqrt{(1 - x^2)} \cdot \sqrt{\left(\dfrac{x + 1}{1 - x} \right)}$

#8: $x \in \text{Real } [-1, 1)$

#9: $\sqrt{(1 - x^2)} \cdot \sqrt{\left(\dfrac{1 + x}{1 - x} \right)} = \dfrac{\sqrt{(1 - x^2)} \cdot \sqrt{(x + 1)}}{\sqrt{(1 - x)}}$

#10: $\sqrt{(1 - x^2)} \cdot \sqrt{\left(\dfrac{1 + x}{1 - x} \right)} - (1 + x) = \dfrac{\sqrt{(1 - x^2)} \cdot \sqrt{(x + 1)}}{\sqrt{(1 - x)}} - x - 1$

#11: $f(x) := \sqrt{(1 - x^2)} \cdot \sqrt{\left(\dfrac{1 + x}{1 - x} \right)}$

#12:   TABLE(f(x), x, -3, 3, 0.5)

It is interesting and strange that the graph of f(x) shows y = x +1 – take care for the gap at x = 1 – but the value table returns quite other function values for f(x)!

That´s the same procedure for the other root-expression:

#13: $g(x) := \sqrt{(x^2 - 1)} \cdot \sqrt{\left(\dfrac{1 + x}{x - 1} \right)}$

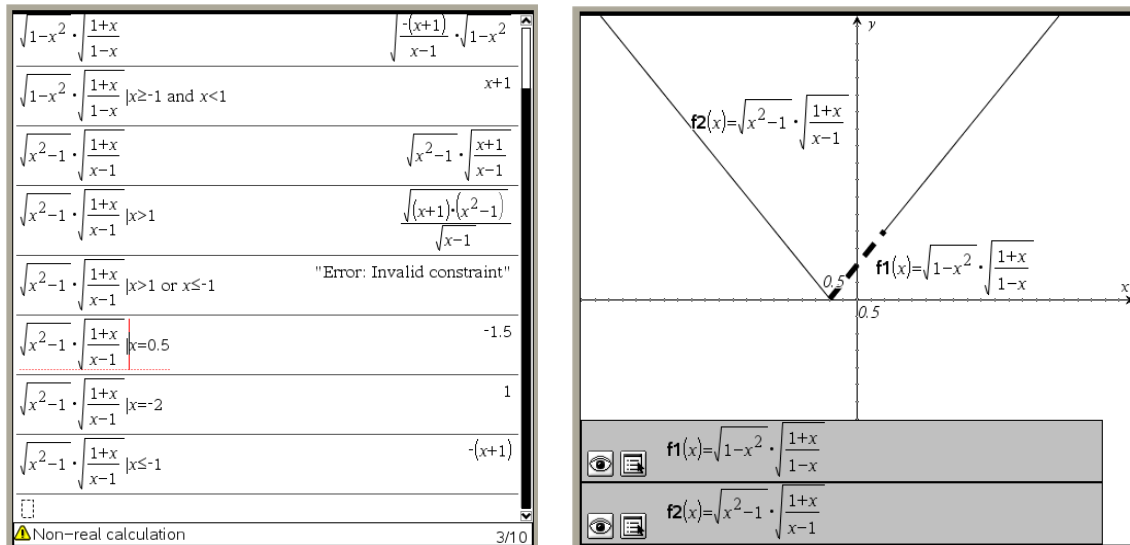#14:   TABLE(g(x), x, -3, 3, 0.5)

My comment of 1995 was: I believe that the plot (DERIVE 3.xx and the value tables) are correct. If you don´t simplify then draw the roots first followed by multiplication. That is the order of performing the arithmetic operations. I hope that my explanation is right and the answer will satisfy you and the readers. Any other comments are highly appreciated. My question about the missing result 0 in expressions #9 (DERIVE 3) and #10 (DERIVE 6) might be answered in this way, too.

How are the answers of the TI-devices? Voyage 200 and TI-NspireCAS behave pretty the same:

$\sqrt{x} \cdot \sqrt{y}$	$\sqrt{x} \cdot \sqrt{y}$
$\sqrt{x} \cdot \sqrt{y} \,\|x{\geq}0 \text{ and } y{\geq}0$	$\sqrt{x{\cdot}y}$
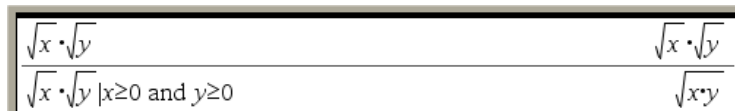
It performs the simplification of the product of the roots.

The next screenshots show how the other expressions are treated by the TIs:

$$\sqrt{1-x^2} \cdot \sqrt{\frac{1+x}{1-x}} \qquad \sqrt{\frac{-(x+1)}{x-1}} \cdot \sqrt{1-x^2}$$

$$\sqrt{1-x^2} \cdot \sqrt{\frac{1+x}{1-x}} \, |x \geq -1 \text{ and } x < 1 \qquad x+1$$

$$\sqrt{x^2-1} \cdot \sqrt{\frac{1+x}{x-1}} \qquad \sqrt{x^2-1} \cdot \sqrt{\frac{x+1}{x-1}}$$

$$\sqrt{x^2-1} \cdot \sqrt{\frac{1+x}{x-1}} \, |x>1 \qquad \frac{\sqrt{(x+1)\cdot(x^2-1)}}{\sqrt{x-1}}$$

$$\sqrt{x^2-1} \cdot \sqrt{\frac{1+x}{x-1}} \, |x>1 \text{ or } x \leq -1 \qquad \text{"Error: Invalid constraint"}$$

$$\sqrt{x^2-1} \cdot \sqrt{\frac{1+x}{x-1}} \, |x=0.5 \qquad -1.5$$

$$\sqrt{x^2-1} \cdot \sqrt{\frac{1+x}{x-1}} \, |x=-2 \qquad 1$$

$$\sqrt{x^2-1} \cdot \sqrt{\frac{1+x}{x-1}} \, |x \leq -1 \qquad -(x+1)$$

⚠ Non-real calculation   3/10

$$f2(x)=\sqrt{x^2-1} \cdot \sqrt{\frac{1+x}{x-1}}$$

$$f1(x)=\sqrt{1-x^2} \cdot \sqrt{\frac{1+x}{1-x}}$$

$$f1(x)=\sqrt{1-x^2} \cdot \sqrt{\frac{1+x}{1-x}}$$

$$f2(x)=\sqrt{x^2-1} \cdot \sqrt{\frac{1+x}{x-1}}$$

Darlene Boharsik, Marietta, Georgia. USA

Thank you for the timely manner in which you forwarded to me all 17 issues of the published DNL.
I am enjoying every copy. I look forward to all future issues. It is quite interesting I find, reading the forum from users around the world.
I am working towards my second career as a math teacher and discovered *DERIVE* 2 years ago.
I feel that there is a need for the US schools to implement a program such as *DERIVE* as an additional tool in the math classroom. I hope to help bring this knowledge to our math departments. Our math department at the Southern College of Technology at this time is a bit slow on change.
Please keep up the good work as I will be spreading the news about *DERIVE's* use in the classroom.

Andreas Zwicknagel, Regensburg, Germany

Lieber Josef, dear members of the DUG!
Schon vom ersten Durchblättern der DNLs bin ich tief beeindruckt. Ich habe gar nicht gewusst oder geahnt, zu was *DERIVE* fähig ist. Zudem bewundere ich das Engagement des Editors. Wir beide haben zwar denselben "Nebenberuf", aber die Zeit zu einem tiefer schürfenden Einstieg in die Programmierfähigkeiten anhand der DNLs wird mir wohl erst in den nächsten Sommerferien bleiben !
Mit freundlichen Grüßen, A.Z.

DNL: I hope you had some time to work with *DERIVE* during the holidays. For you and all our members I add one plot which I have received from a young and very enthusiastic Belgian *DERIVE*-math teacher. I met him at Turnhout. where I had a workshop about graphics with *DERIVE*. He told that his students –s econdary school - use *DERIVE* to design phantasy buildings. He promised to send some prints. Dear Luk Van den Broeck, thank you very much for not only promising, but promptly sending. Congratulations to your students. You can find other spread in this issue.
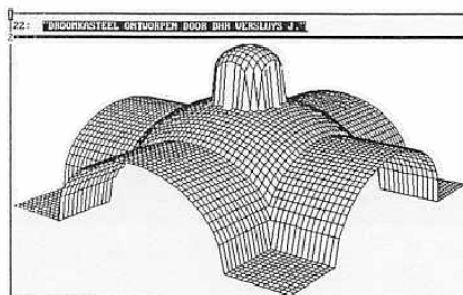
Didactics of CAS
Conic Sections Escape R^2

Dr. Karl Fuchs, ACDCA, Salzburg, Vienna

0 Didactical Introduction

In the 7th class of the Austrian Grammar school we have to teach the conic sections: ellipse, hyperbola and parabola. Although we have the methods of analytical geometry to create the equations of conic sections in R^3 they have no chance to escape the R^2 in the moment. We are discussing focus-characteristics of conic sections:

$$\overline{XF_1} + \overline{XF_2} = 2a; \ 2a > \overline{F_1F_2} \text{ and } e^2 = a^2 - b^2 \qquad \text{for the ellipse,}$$

$$\left| \overline{XF_1} + \overline{XF_2} \right| = 2a; \ 2a < \overline{F_1F_2} \text{ and } e^2 = a^2 + b^2 \qquad \text{for the hyperbola,}$$

$$\overline{Xl} = \overline{XF} \qquad\qquad\qquad\qquad\qquad\qquad \text{for the parabola,}$$

derive the equations:

$$b^2x^2 + a^2x^2 = a^2b^2, \ b^2x^2 - a^2x^2 = a^2b^2 \ \text{ and } \ y^2 = 2px,$$

intersect the curves and even intersect them with straight lines. But although we are doing such a lot of things with the curves the students ask themselves: "Why are these curves called **conic** sections?" Now the teacher has the possibility to refer the student to a picture in the textbook that shows a cone intersected by different planes generating the discussed curves and the teacher can continue telling them that this is the reason why these curves are called conic curves. But most of our students – especially nowadays – aren't interested in **telling**, they are interested in **doing** things – even mathematical things.

In 1983 Univ.Prof. Dr. Fritz Schweiger, Professor for the Didactics of Mathematics at the University of Salzburg wrote a didactical article dealing with the problem of teaching conic sections in Grammar school. In an appendix he suggested to escape to R^3 to make these curves interesting for students with high ability in mathematics. At that time we were able to do the analytical geometry in mathematics, the illustration of the processes and results we had to pass on to the teacher in geometry with regard to construction which is an independent subject called 'Darstellende Geometrie' in Austrian Grammar schools. This was really a problem.

I think nowadays with *DERIVE* we have an instrument to do these illustrations, that we better call visualizations now because of the dynamic aspect of drawing on the computer screen. "*DERIVE* isn't good at plotting 3D-objects" is an argument I often hear from several different users of *DERIVE* – especially from teachers. Certainly on one hand we don't want *DERIVE* to go the way of the huge CAS-systems like *MAPLE* or *MATHEMATICA* in graphics, which are too big for the school but there is a lot we can do with *DERIVE* – even with 3D-objects.
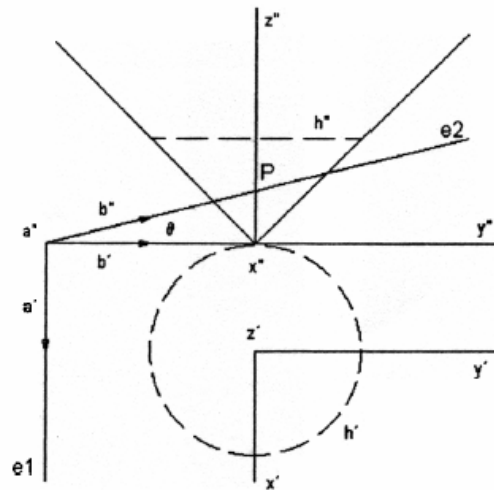
1 The Problem

The two objects of our investigations are

(1)   **cone** *k* with a right angle at the top
         $k : x^2 + y^2 = z^2$ and

(2)   an intersecting **plane** *e* containing point
         $P(0,0,z_1)$, parallel to the *x*-axis with an

         angle θ of inclination

$$e : \begin{cases} x = u \\ y = v \cdot \cos\theta \\ z = z_1 + v \cdot \sin\theta \end{cases},$$

         $a = (1,0,0) \perp b = (0, \cos\theta, \sin\theta)$

The figure shows the two objects in a front view and a top view (horizontal and frontal normal projection).

Let´s find the general form of the curve in plane *e* intersecting the cone *k* and the plane *e*.

$k \cap e$:

$$u^2 + v^2 \cos^2\theta = z_1^2 + 2z_1 v \sin\theta + v^2 \sin^2\theta$$
$$u^2 + v^2 (\cos^2\theta - \sin^2\theta) - 2z_1 v\sin\theta = z_1^2$$
$$u^2 + v^2 \cos 2\theta - 2z_1 v\sin\theta = z_1^2$$

The result is the equation of a curve in plane *e*. I decided to perform this sequence by hand to show the students that there is no miracle in manipulating the equation (elementary algebraic transformation and the trig identity $\cos^2\theta - \sin^2\theta = \cos 2\theta$) in order to obtain the general form of the curve in *e*.

1 The *DERIVE* Session

Now we are going to visualize the curves for **different values of θ** (using functions of utility file `GraphicsFunctions.mth`). We choose $z_1 = 0.5$ for the following examples.

1st step: Defining the curve and deriving the functional form $u = f(v, \theta)$ for plotting.

```
#1:    LOAD(C:\Programme\TI Education\Derive 6\Math\GraphicsFunctions.mth)

        2    2                     2
#2:    u  + v ·COS(2·θ) − v·SIN(θ) = 0.5

           2    2                     2
#3:    SOLVE(u  + v ·COS(2·θ) − v·SIN(θ) = 0.5 , u)

              2
        √(− 4·v ·COS(2·θ) + 4·v·SIN(θ) + 1)
#4:    u = − ─────────────────────────────────── ∨ u =
                           2

           2
        √(− 4·v ·COS(2·θ) + 4·v·SIN(θ) + 1)
        ───────────────────────────────────
                       2
```

Our solution vector will have the form $[u, v \cos(\theta), z_1 + v \sin(\theta)]$.

2nd step: Finding the curve for $\theta = 0$.

2.1   Substitute 0 for θ in #4 and simplify:

#5: $u = -\dfrac{\sqrt{(-4 \cdot v^2 \cdot \cos(2 \cdot 0) + 4 \cdot v \cdot \sin(0) + 1)}}{2} \quad \lor \quad u =$

$\dfrac{\sqrt{(-4 \cdot v^2 \cdot \cos(2 \cdot 0) + 4 \cdot v \cdot \sin(0) + 1)}}{2}$

#6: $u = -\dfrac{\sqrt{(1 - 4 \cdot v^2)}}{2} \quad \lor \quad u = \dfrac{\sqrt{(1 - 4 \cdot v^2)}}{2}$

2.2   Create vectors k_1 and k_2 as solution vectors:

#7: $k1 := \left[\dfrac{\sqrt{(1 - 4 \cdot v^2)}}{2}, \ v \cdot \cos(0), \ 0.5 + v \cdot \sin(0) \right]$

#8: $k2 := \left[-\dfrac{\sqrt{(1 - 4 \cdot v^2)}}{2}, \ v \cdot \cos(0), \ 0.5 + v \cdot \sin(0) \right]$

3rd step Call the axes (contained in the preloaded utility file) and apply the isometric projection on k_1 and k_2:

#9:   axes

#10:   [ISOMETRIC(k1), ISOMETRIC(k2)]

4th step: Plot #9 with the parameter between -1 and +1.

Plot #10 with the parameter $-0.5 \leq v \leq +0.5$ (Take care for the domain – expr #7 & #8!!)

It might be a good question for students, why $-\pi \leq v \leq +\pi$ does not result in a complete plot!
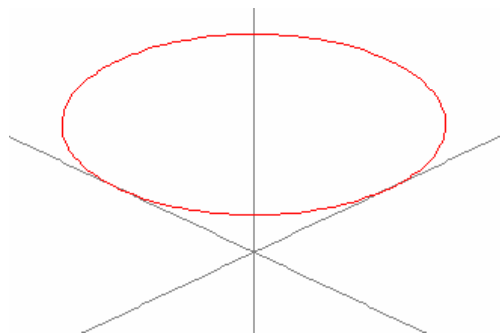
We have the isometric picture of a circle together with the axes.

We can generate the other curves in a very similar way:
(I redefine the general form of the intersection curves.)

#12: $k\_1 := \left[-\dfrac{\sqrt{(-4 \cdot v^2 \cdot COS(2 \cdot \theta) + 4 \cdot v \cdot SIN(\theta) + 1)}}{2}, \; v \cdot COS(\theta), \; v \cdot SIN(\theta) + \dfrac{1}{2} \right]$

#13: $k\_2 := \left[\dfrac{\sqrt{(-4 \cdot v^2 \cdot COS(2 \cdot \theta) + 4 \cdot v \cdot SIN(\theta) + 1)}}{2}, \; v \cdot COS(\theta), \; v \cdot SIN(\theta) + \dfrac{1}{2} \right]$

We obtain the intersection as an ellipse for $\theta = \dfrac{\pi}{6}$:

#14: $SUBST\left([ISOMETRIC(k\_1), ISOMETRIC(k\_2)], \; \theta, \; \dfrac{\pi}{6} \right)$

#15: $\left[\begin{array}{l} \dfrac{\sqrt{(-2 \cdot v^2 + 2 \cdot v + 1)}}{2} + \dfrac{\sqrt{3} \cdot v}{2} \quad -\dfrac{\sqrt{(-2 \cdot v^2 + 2 \cdot v + 1)}}{4} + v \cdot \left(\dfrac{1}{2} - \dfrac{\sqrt{3}}{4} \right) + \dfrac{1}{2} \\[4mm] \dfrac{\sqrt{3} \cdot v}{2} - \dfrac{\sqrt{(-2 \cdot v^2 + 2 \cdot v + 1)}}{2} \quad -\dfrac{\sqrt{(-2 \cdot v^2 + 2 \cdot v + 1)}}{4} + v \cdot \left(\dfrac{1}{2} - \dfrac{\sqrt{3}}{4} \right) + \dfrac{1}{2} \end{array} \right]$

#16: $SOLVE(-2 \cdot v^2 + 2 \cdot v + 1 \geq 0, \; v) = \dfrac{1}{2} - \dfrac{\sqrt{3}}{2} \leq v \leq \dfrac{\sqrt{3}}{2} + \dfrac{1}{2}$

#17: $axes$
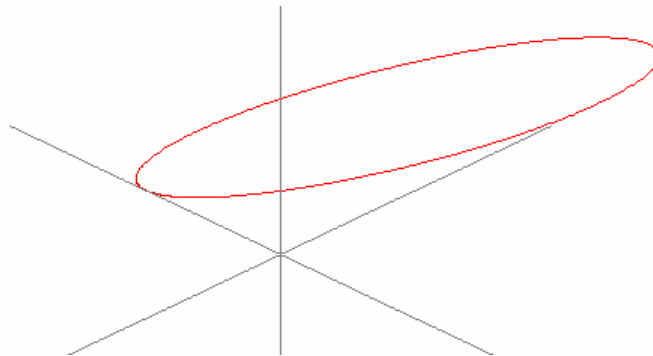
The solutions of inequality #16 are needed for entering the boundaries of the parameter v when plotting #15. It might be a good question for students why plots generated with larger ranges for v, eg $-5 \leq v \leq +5$ show gaps in the plots!! (Try it! The square root must be entered as sqrt(..).)

We obtain the intersection as a parabola for $\theta = \dfrac{\pi}{4}$. Why?

#18: $SUBST\left([ISOMETRIC(k\_1), ISOMETRIC(k\_2)], \; \theta, \; \dfrac{\pi}{4} \right)$

#19: $\left[\begin{array}{l} \dfrac{\sqrt{(2 \cdot \sqrt{2} \cdot v + 1)}}{2} + \dfrac{\sqrt{2} \cdot v}{2} \quad -\dfrac{\sqrt{(2 \cdot \sqrt{2} \cdot v + 1)}}{4} + \dfrac{\sqrt{2} \cdot v}{4} + \dfrac{1}{2} \\[4mm] \dfrac{\sqrt{2} \cdot v}{2} - \dfrac{\sqrt{(2 \cdot \sqrt{2} \cdot v + 1)}}{2} \quad -\dfrac{\sqrt{(2 \cdot \sqrt{2} \cdot v + 1)}}{4} + \dfrac{\sqrt{2} \cdot v}{4} + \dfrac{1}{2} \end{array} \right]$
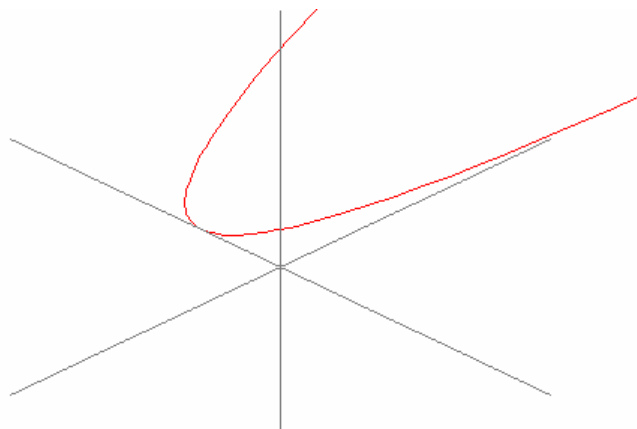
#20: $SOLVE(2 \cdot \sqrt{2} \cdot v + 1 \geq 0, \; v) = v \geq -\dfrac{\sqrt{2}}{4}$
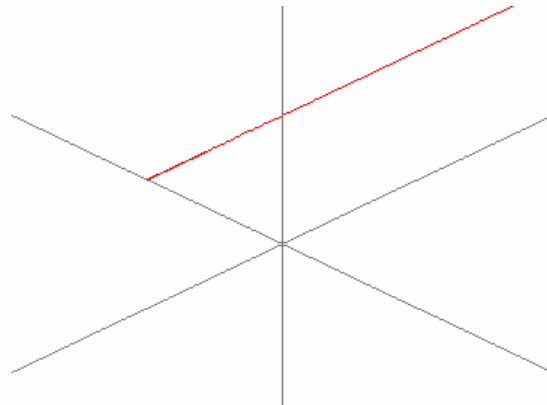
The plot isn´t very attracting und you cannot really see the parabola. To obtain a better visualization the object was rotated applying ROTATE_Z(π/4) – which results in a 90°-turn. (Take care for transposing the k_1- and k_2-vectors before multiplying with the rotation matrix!)

#21: SUBST$\left(\left[\text{ISOMETRIC}\left(\text{ROTATE\_Z}\left(\dfrac{\pi}{4}\right)\cdot[\text{k\_1}]'\right),\ \text{ISOMETRIC}\left(\text{ROTATE\_Z}\left(\dfrac{\pi}{4}\right)\cdot[\text{k\_2}]'\right)\right],\ \theta,\ \dfrac{\pi}{4}\right)$

#22:
$$\begin{bmatrix} \left[v,\ \dfrac{\sqrt{2}\cdot\sqrt{(2\cdot\sqrt{2}\cdot v + 1)}}{4} + \dfrac{\sqrt{2}\cdot v}{2} + \dfrac{1}{2}\right] \\ \left[v,\ -\dfrac{\sqrt{2}\cdot\sqrt{(2\cdot\sqrt{2}\cdot v + 1)}}{4} + \dfrac{\sqrt{2}\cdot v}{2} + \dfrac{1}{2}\right] \end{bmatrix}$$

The hyperbola is missing, let´s take $\theta = \dfrac{\pi}{3}$.

#23: SUBST$\left([\text{ISOMETRIC}(\text{k\_1}),\ \text{ISOMETRIC}(\text{k\_2})],\ \theta,\ \dfrac{\pi}{3}\right)$

#24:
$$\begin{bmatrix} \dfrac{\sqrt{(2\cdot v^2 + 2\cdot\sqrt{3}\cdot v + 1)}}{2} + \dfrac{v}{2} & \dfrac{\sqrt{(2\cdot v^2 + 2\cdot\sqrt{3}\cdot v + 1)}}{4} + v\cdot\left(\dfrac{\sqrt{3}}{2} - \dfrac{1}{4}\right) + \dfrac{1}{2} \\ \dfrac{v}{2} - \dfrac{\sqrt{(2\cdot v^2 + 2\cdot\sqrt{3}\cdot v + 1)}}{2} & -\dfrac{\sqrt{(2\cdot v^2 + 2\cdot\sqrt{3}\cdot v + 1)}}{4} + v\cdot\left(\dfrac{\sqrt{3}}{2} - \dfrac{1}{4}\right) + \dfrac{1}{2} \end{bmatrix}$$

#25: SOLVE$(2\cdot v^2 + 2\cdot\sqrt{3}\cdot v + 1 \geq 0,\ v) = \left(v \leq -\dfrac{\sqrt{3}}{2} - \dfrac{1}{2} \lor v \geq \dfrac{1}{2} - \dfrac{\sqrt{3}}{2}\right)$
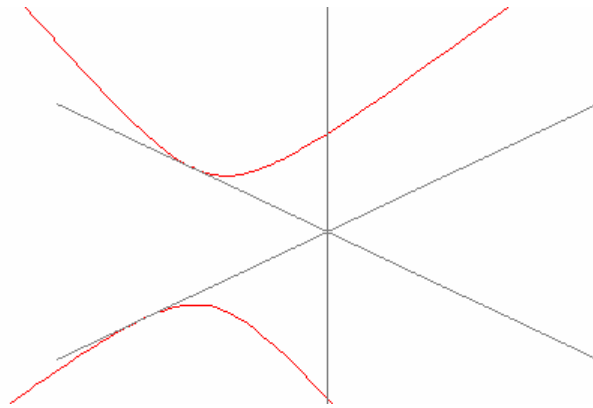
One case is left. Intersecting the cone with a plane through its vertex (with $\theta = \dfrac{\pi}{2}$). The plane is verti-cal now. The intersection curve is a pair of straight lines (real or complex).

#26: SUBST$\left([\text{ISOMETRIC}(k\_1), \text{ISOMETRIC}(k\_2)], \theta, \dfrac{\pi}{2}\right)$

#27:
$$\begin{bmatrix} \dfrac{|2 \cdot v + 1|}{2} & \dfrac{|2 \cdot v + 1|}{4} + v + \dfrac{1}{2} \\[3mm] -\dfrac{|2 \cdot v + 1|}{2} & -\dfrac{|2 \cdot v + 1|}{4} + v + \dfrac{1}{2} \end{bmatrix}$$
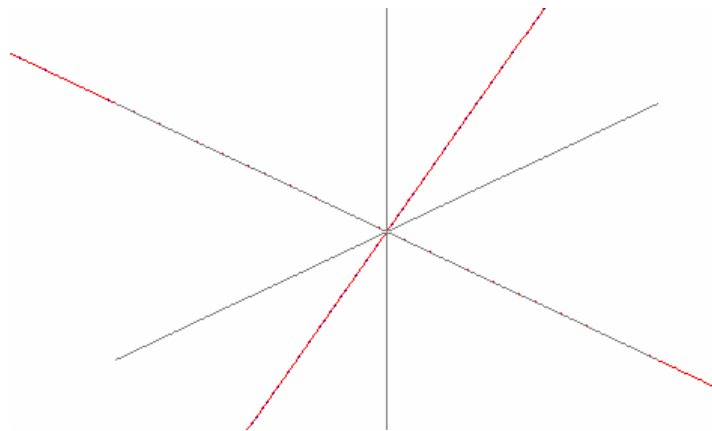
Here again the plot is not very informative.
I took another plane through the vertex ($z_1 = 0$ and $\theta = 3\pi/8$):

#28: $u^2 + v^2 \cdot \text{COS}(2 \cdot \theta) = 0$

#29: SOLVE($u^2 + v^2 \cdot \text{COS}(2 \cdot \theta) = 0$, u)

#30: $u = -|v| \cdot \sqrt{(-\text{COS}(2 \cdot \theta))} \ \vee \ u = |v| \cdot \sqrt{(-\text{COS}(2 \cdot \theta))}$

#31: co1 := $[-|v| \cdot \sqrt{(-\text{COS}(2 \cdot \theta))}, v \cdot \text{COS}(\theta), v \cdot \text{SIN}(\theta)]$

#32: co2 := $[|v| \cdot \sqrt{(-\text{COS}(2 \cdot \theta))}, v \cdot \text{COS}(\theta), v \cdot \text{SIN}(\theta)]$

#33: SUBST$\left([\text{ISOMETRIC}(co1), \text{ISOMETRIC}(co2)], \theta, \dfrac{3 \cdot \pi}{8}\right)$

#34:
$$\begin{bmatrix} \dfrac{2^{3/4}\cdot|v|}{2} + \dfrac{v\cdot\sqrt{(2-\sqrt{2})}}{2} & \dfrac{2^{3/4}\cdot|v|}{4} + \dfrac{v\cdot\sqrt{(10-\sqrt{2})}}{4} \\ \dfrac{v\cdot\sqrt{(2-\sqrt{2})}}{2} - \dfrac{2^{3/4}\cdot|v|}{2} & \dfrac{v\cdot\sqrt{(10-\sqrt{2})}}{4} - \dfrac{2^{3/4}\cdot|v|}{4} \end{bmatrix}$$
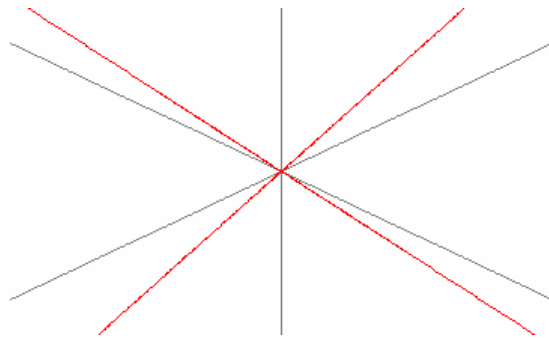
References

Schweiger Fritz, Analytische Geometrie oder sind die Kegelschnitte noch zu retten?
Mathematik im Unterricht, Nr. 7, 27 – 29, 1983

# 3D plots of Karl´s Conic Sections

Josef Böhm, Würmla, Austria

In July I was invited by Jan Vermeylen to give a presentation and to hold a workshop at the Conference of the VVWL (Vlaamse Vereniging Wiskunde Leraaren = Association of Flemish Mathematics Teachers) at Turnhout, Belgium. Jan gave a workshop "*Conic Sections*" and asked to produce 3D-pictures of the curves. I tried and I add the plots to Karl´s examples from the contribution above using my results.

These are the results with Karl´s special assumptions. In the following you can find a more generalized solution of the problem. I must admit that it took some time to find appropriate values for the eye-coordinates in order to obtain nice looking pictures.

The Circle
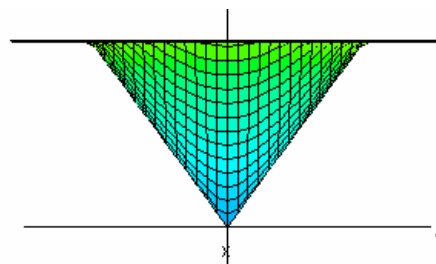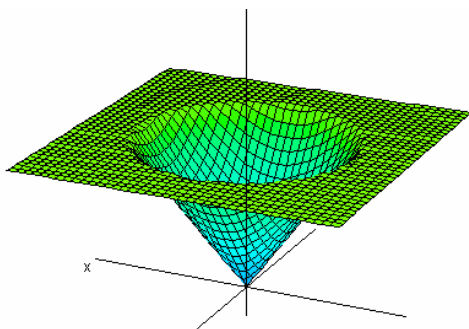
The Ellipse
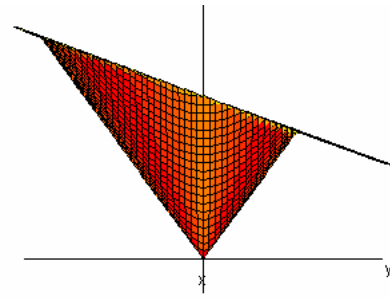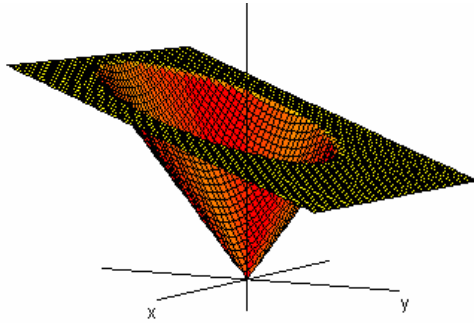
The Parabola
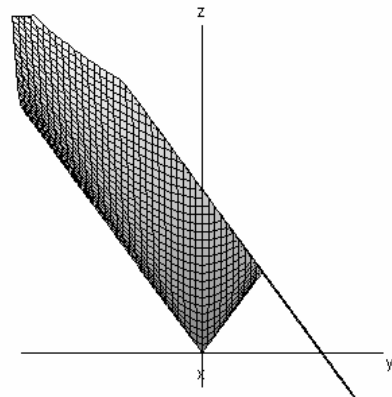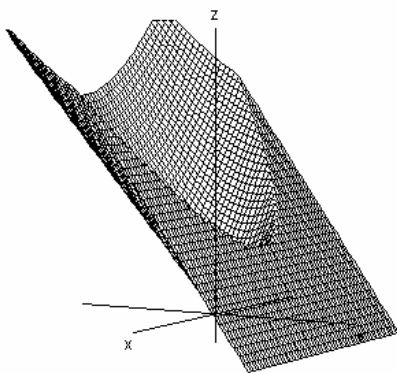
The Hyperbola
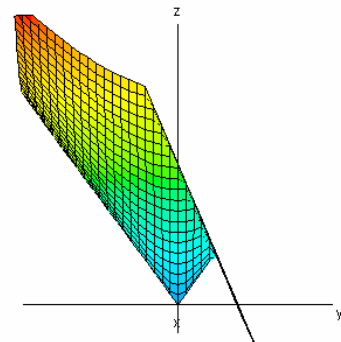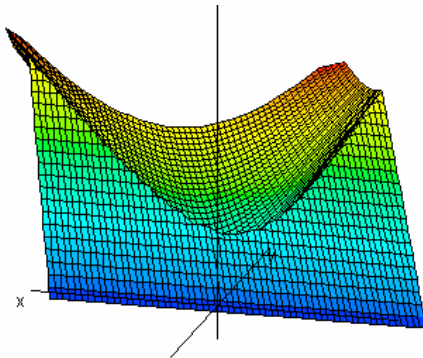
A Pair of Lines
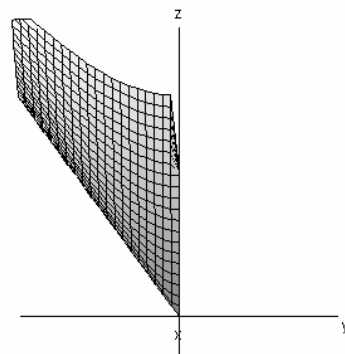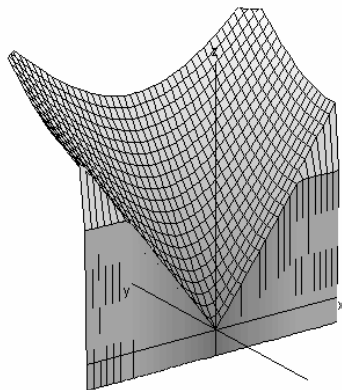
I define the plane *E* by its intercepts on the axes and the cone:

#1: $E(a, b, c) := \lim_{c\_\to c} \ \lim_{b\_\to b} \ \lim_{a\_\to a} \ -\dfrac{c\_\cdot(b\_\cdot x + a\_\cdot(y - b\_))}{a\_\cdot b\_}$

#2: $con := \sqrt{(x^2 + y^2)}$

#3: $COSEC(\theta, z1) := MIN\left(con, \ E\left(\infty, \ \dfrac{z1}{TAN(\theta)}, \ z1\right)\right)$

#4: $COSEC(0, 0.5)$

#5: $\dfrac{2\cdot\sqrt{(x^2 + y^2)} + 1}{4} - \dfrac{\left|2\cdot\sqrt{(x^2 + y^2)} - 1\right|}{4}$

The settings for the plot box (circle) are:

	Minimum	Maximum	Scale
x:	-0.75	0.75	0.75
y:	-0.75	0.75	0.75
z:	-0.1	1	0.55

#6: $COSEC\left(\dfrac{\pi}{6}, \ 0.5\right)$

#7: $COSEC\left(\dfrac{\pi}{4}, \ 0.5\right)$

#8: $COSEC\left(\dfrac{\pi}{3}, \ 0.5\right)$
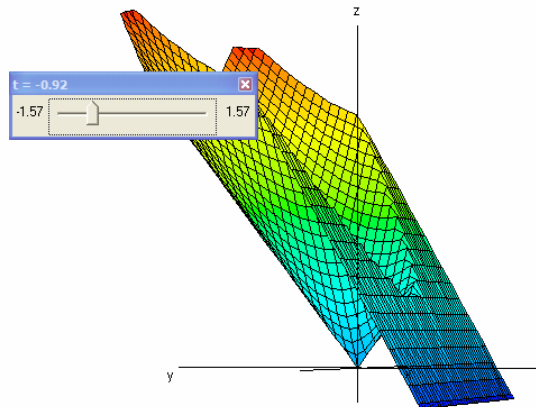
#9: $COSEC\left(\dfrac{\pi}{2}, \ 0.5\right) = ?$

#10: $COSEC(1.57, 0.5)$

#11: $MIN\left(con, \ 0\cdot x + TAN\left(\dfrac{3\cdot\pi}{8}\right)\cdot y\right)$

In times of Derive 6 we introduce a slider bar for parameter *t* with $-\dfrac{\pi}{2} \leq t \leq \dfrac{\pi}{2}$.
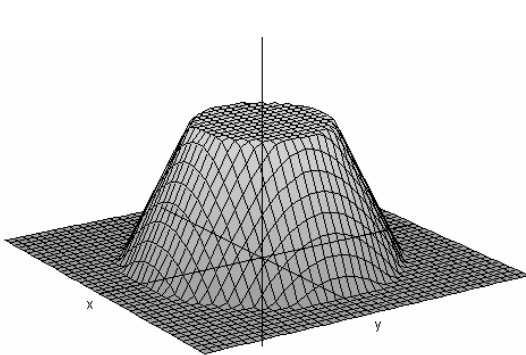
#12: $COSEC(t, 0.5)$

The more generalized version: Find the intersection curve between the *cone0(h,r)* and the plane *E(a,b,c)* with its intercepts on the axes *a*, *b* and *c*.
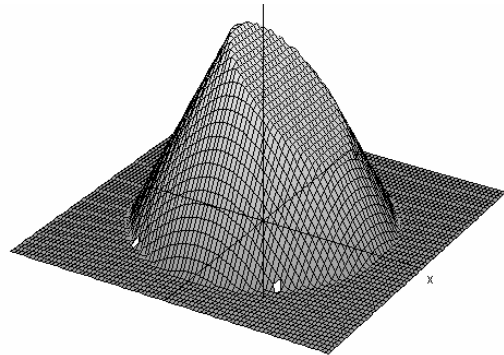
#1: $\displaystyle \text{CONE0}(h, r) := \frac{r \cdot h - \sqrt{(x^2 + y^2)} \cdot h}{r}$

#2: $\displaystyle E(a, b, c) := \lim_{c\_ \to c} \lim_{b\_ \to b} \lim_{a\_ \to a} - \frac{c\_ \cdot (b\_ \cdot x + a\_ \cdot (y - b\_))}{a\_ \cdot b\_}$
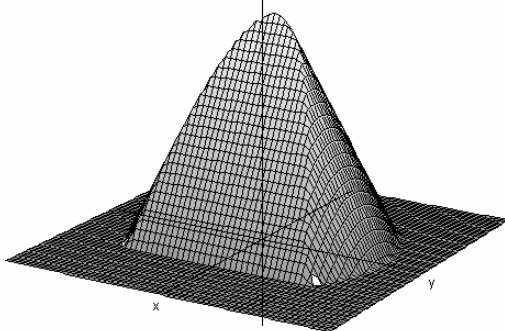
#3: $\text{CONESEC}(h, r, a, b, c) := \text{MAX}(\text{MIN}(\text{CONE0}(h, r), E(a, b, c)), 0)$

#4:   CONESEC(6, 3, ∞, ∞, 3)

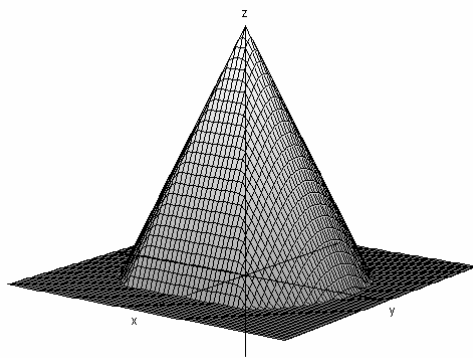
#5:   CONESEC(6, 3, 4, ∞, 4)
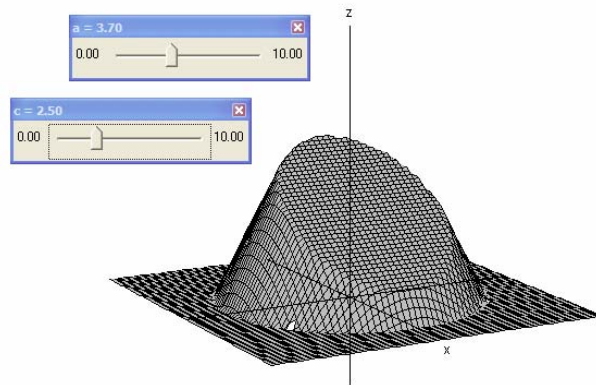

#6:   CONESEC(6, 3, 2, ∞, 4)
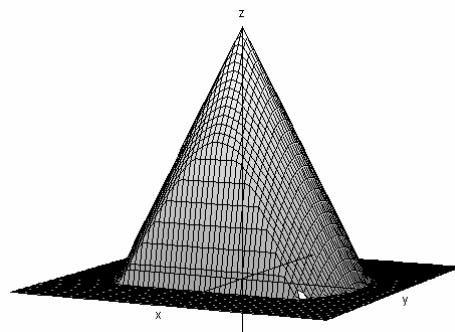
#7:   CONESEC(6, 3, 2, ∞, 10)

#8:   CONESEC(6, 3, 2.5, ∞, 6)   #9: CONESEC(6, 3, a, ∞, c)

Cubic Splines Part 2 & 3

Otto Reichel, St. Pölten & Leo Klingen, Bonn

In the second part of the contribution dealing with Cubic Splines you can see how Otto Reichel solved the task. There are some similarities with my approach. We talked about and very soon we found the reason for this fact: we used the same books about computer graphics. Nevertheless I think that it could be interesting to compare and to find the differences. I am sure that you can follow the 'program' without further explanations if you use the theoretic basics about Cubic Splines (1). Otto used in his files the same nodes as I did. Then he added a more extended example. I add the plot.

```
#1:   Cubic Splines by O. Reichel; RSPLI.MTH
```

$$\#2: \quad H(a) := \text{VECTOR}(a_{k+1,1} - a_{k,1}, k, \text{DIM}(a) - 1)$$

```
#3:   H1(a) := APPEND([0], H(a))
```

$$\#4: \quad F(a) := \text{VECTOR}(2\cdot((H(a))_k + (H(a))_{k-1}), k, 2, \text{DIM}(H(a)))$$

```
#5:   F1(a) := APPEND([0], F(a))
```

$$\#6: \quad G(a) := \text{APPEND}\left([0], \text{VECTOR}\left(3\cdot\left(\frac{a_{k+1,2} - a_{k,2}}{(H(a))_k} - \frac{a_{k,2} - a_{k-1,2}}{(H(a))_{k-1}}\right), k, 2, \text{DIM}(a) - 1\right)\right)$$

$$\#7: \quad \text{MAT1}(a) := \text{VECTOR}(\text{VECTOR}(\text{IF}(i\cdot j = 1, 1, \text{IF}(i = j, (F1(a))_i, \text{IF}(i = 1 \vee j = 1, 0, \text{IF}(|i - j| = 1, (H1(a))_{\text{MAX}(i, j)}, 0)))), i, \text{DIM}(a) - 1), j, \text{DIM}(a) - 1)$$

$$\#8: \quad \text{CK}(a) := \text{MAT1}(a)^{-1} \cdot G(a)$$

#9: $AK(a) := VECTOR(a_{k,2}, k, DIM(a) - 1)$

#10: $c1 := APPEND(c, [0])$

#11: $A1(a) := a'_2$

#12: $BK(a) := VECTOR\left(\dfrac{(A1(a))_{k+1} - (A1(a))_k}{(H(a))_k} - \dfrac{(H(a))_k \cdot (c1_{k+1} + 2 \cdot c1_k)}{3}, k,\right.$

$\left. DIM(a) - 1\right)$

#13: $DK(a) := VECTOR\left(\dfrac{c1_{k+1} - c1_k}{3 \cdot (H(a))_k}, k, DIM(a) - 1\right)$

#14: $PK(a) := VECTOR\left((AK(a))_k + (BK(a))_k \cdot (x - a_{k,1}) + c_k \cdot (x - a_{k,1})^2 + (DK(a))_k \cdot (x - a_{k,1})^3, k, DIM(a) - 1\right)$

#15: $STK(a) := VECTOR(\chi(a_{k,1}, x, a_{k+1,1}), k, DIM(a) - 1)$

#16: $KUBSPLI(a) := PK(a) \cdot STK(a)$

#17: $nodes := \begin{bmatrix} 0 & 1 \\ 1 & 2 \\ 3 & 4 \\ 4 & 2.5 \\ 5 & 0 \\ 6 & 1 \end{bmatrix}$
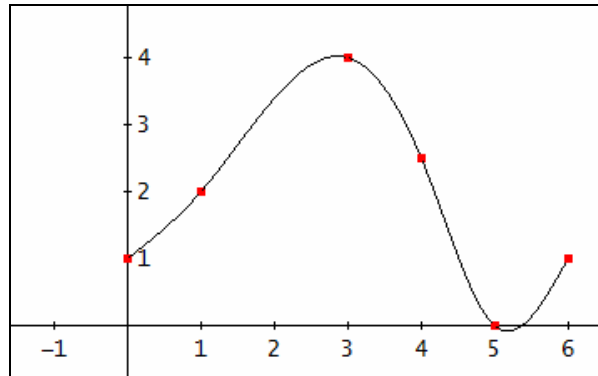
#18: $c := CK(nodes)$

#19: $c := \left[0, \dfrac{15}{38}, -\dfrac{45}{38}, -\dfrac{45}{38}, \dfrac{111}{38}\right]$
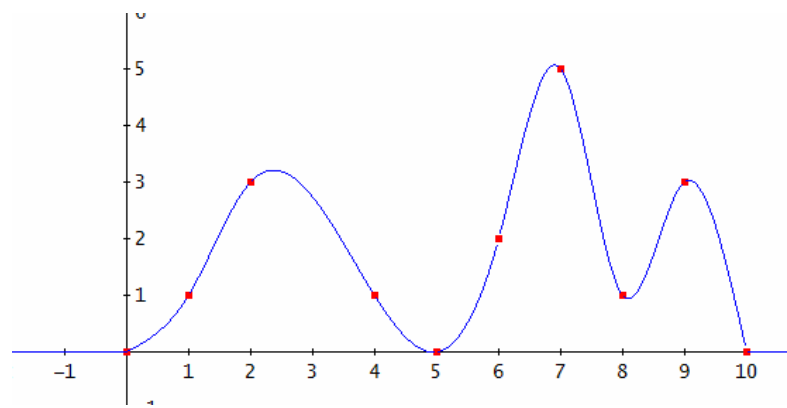
#20: $KUBSPLI(nodes)$

#21: $\dfrac{(x - 5) \cdot (37 \cdot x^2 - 481 \cdot x + 1516) \cdot SIGN(x - 6)}{76} -$

$\dfrac{89 \cdot (x^3 - 15 \cdot x^2 + 75 \cdot x - 125) \cdot SIGN(x - 5)}{76} + \ldots$

You can immediately plot expression #20; don´t forget to activate Options > Approximate before plotting in the 2D-Plot Window.

$$\text{\#22:}\quad example2 := \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 2 & 3 \\ 4 & 1 \\ 5 & 0 \\ 6 & 2 \\ 7 & 5 \\ 8 & 1 \\ 9 & 3 \\ 10 & 0 \end{bmatrix}$$

Perform the same procedure as above, leading to KUBSPLI(example2):

You have to define the result of CK(example2) as c:= and then plot KUBSPLI(example2).

Leo Klingen´s Splines

Leo Klingen wrote the most compact DERIVE-program fort he Cubic Splines. As a special you will find a function for parametric splines, too. Additionally there is another function to approximate any given function by cubic splines.

(Even in 1995 some problems occurred with this file changing from one DERIVE-version to the other. I adapted the file for DERIVE 6.10. Josef)

#1:   Notation := Decimal

#2: $\left[\text{SPLINE\_A1}(h, z) := h_{z-1}, \quad \text{SPLINE\_B1}(h, z) := 2 \cdot (h_{z-1} + h_z) \right]$

#3: $\text{SPLINE\_C1}(h, z) := h_z$

#4: $\text{SPLINE\_D1}(a, h, z) := 3 \cdot \left(\dfrac{a_{z+1} - a_z}{h_z} - \dfrac{a_z - a_{z-1}}{h_{z-1}} \right)$

#5: $\text{SPLINE\_D}(c, h, n) := \text{VECTOR}\left(\dfrac{c_{a\_+1} - c_{a\_}}{3 \cdot h_{a\_}}, a\_, n-1 \right)$

#6: $\text{SPLINE\_B}(a, c, h, n) := \text{VECTOR}\left(\dfrac{\dfrac{a_{b_- + 1} - a_{b_-}}{h_{b_-}} - \dfrac{h_{b_-} \cdot (2 \cdot c_{b_-} + c_{b_- + 1})}{3}}{}, b_-, n - 1 \right)$

#7: $\text{SPLINE\_AUX}(m, a, c, n, h, x) := \text{VECTOR}\left(\left(a_{c_-} + (\text{SPLINE\_B}(a, c, h, n))_{c_-} \cdot (x - m_{c_-,1}) + c_{c_-} \cdot (x - m_{c_-,1})^2 + (\text{SPLINE\_D}(c, h, n))_{c_-} \cdot (x - \right.\right.$

$\left.\left. m_{c_-,1})^3 \right) \cdot \chi(m_{c_-,1}, x, m_{c_- + 1,1}), c_-, n - 1 \right)$

#8: $\text{SPLINE\_AUX2}(m, n, x) := \text{SPLINE\_AUX}(m, m'_2, \text{ELEMENT}(\text{ROW\_REDUCE}(\text{APPEND}(\text{APPEND}([\text{VECTOR}(\text{IF}(s_- = 1, 1, 0), s_-, n)], \text{VECTOR}(\text{VECTOR}(\text{IF}(s_- = z_-$

$- 1, \text{SPLINE\_A1}(\text{APPEND}(\text{VECTOR}(m'_{1,i_- + 1} - m'_{1,i_-}, i_-, n - 1), [0]), z_-), \text{IF}(s_- = z_-, \text{SPLINE\_B1}(\text{APPEND}(\text{VECTOR}(m'_{1,i_- + 1} - m'_{1,i_-},$

$i_-, n - 1), [0]), z_-), \text{IF}(s_- = z_- + 1, \text{SPLINE\_C1}(\text{APPEND}(\text{VECTOR}(m'_{1,i_- + 1} - m'_{1,i_-}, i_-, n - 1), [0]), z_-), 0))), s_-, n), z_-, 2, n -$

$1)), [\text{VECTOR}(\text{IF}(s_- = n, 1, 0), s_-, n)]), \text{APPEND}([[0]], \text{APPEND}([\text{VECTOR}(\text{SPLINE\_D1}(m'_2, \text{APPEND}(\text{VECTOR}(m'_{1,i_- + 1} - m'_{1,i_-}, i_-, n - 1),$

$[0]), z_-), z_-, 2, n - 1)]', [[0]])))', n + 1), n, \text{APPEND}(\text{VECTOR}(m'_{1,i_- + 1} - m'_{1,i_-}, i_-, n - 1), [0]), x)$

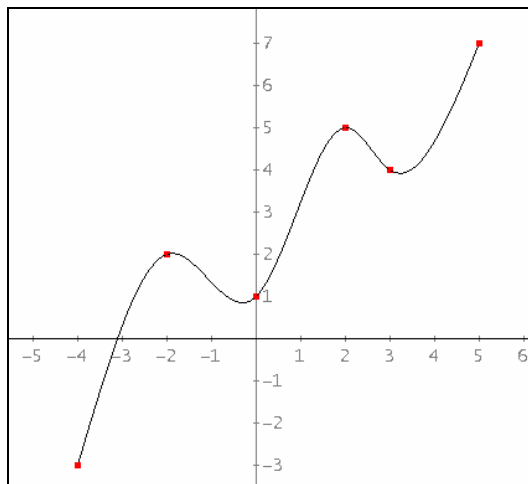#9: $\text{SPLINE}(m, x) := \text{SPLINE\_AUX2}(m, \text{DIMENSION}(m), x)$

#10: $\text{SPLINE\_INTERPOLATION}(f, x, a) := \text{SPLINE}(\text{VECTOR}([a_{k_-}, \lim_{x \to a \downarrow k_-} f], k_-, 1, \text{DIMENSION}(a)), x)$

#11: $\text{SPLINE}\left(\begin{bmatrix} -4 & -3 \\ -2 & 2 \\ 0 & 1 \\ 2 & 5 \\ 3 & 4 \\ 5 & 7 \end{bmatrix}, x\right)$

#12: $\left[1.31027 \cdot 10^{-4} \cdot (1025 \cdot x^3 + 12300 \cdot x^2 + 35560 \cdot x + 22488) \cdot \text{SIGN}(x + 2) - 1.31027 \cdot 10^{-4} \cdot (1025 \cdot x^3 + 12300 \cdot x^2 + 35560 \cdot x + 22488) \cdot \text{SIGN}(x + 4), 1.31 \right.$

#12 expanded:

#13: $\left[-0.134302675 \cdot x^3 \cdot \text{SIGN}(x + 4) - 1.6116321 \cdot x^2 \cdot \text{SIGN}(x + 4) - 4.65932012 \cdot x \cdot \text{SIGN}(x + 4) - 2.946535176 \cdot \text{SIGN}(x + 4) + 0.134302675 \cdot x^3 \cdot \text{SIGN}(x + \right.$

#14: $\text{SPLINE\_INTERPOLATION}\left(\text{SIN}(x), x, \left[-\pi, -\dfrac{\pi}{2}, 0, \dfrac{\pi}{2}, \pi \right]\right)$

#15: $\left[1.25926 \cdot 10^{-10} \cdot (4.53299 \cdot 10^6 \cdot x^2 + 2.84816 \cdot 10^7 \cdot x + 1.11847 \cdot 10^7) \cdot |113 \cdot x + 355| - 1.25926 \cdot 10^{-10} \cdot (113 \cdot x + 355) \cdot (4.53299 \cdot 10^6 \cdot x^2 + 2.84816 \cdot 10^7 \cdot \right.$

#16: $\left[-0.06450280675 \cdot x^3 \cdot \text{SIGN}(226 \cdot x + 355) - 0.6079244187 \cdot x^2 \cdot \text{SIGN}(226 \cdot x + 355) - 1.432387988 \cdot x \cdot \text{SIGN}(226 \cdot x + 355) - 0.4999978089 \cdot \text{SIGN}(226 \cdot x + \right.$
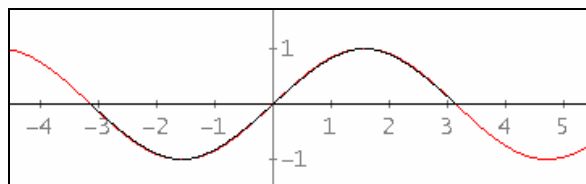
#17: 5 example-points for parametric plotting

$$
#18: \quad p := \begin{bmatrix} 1 & 0 \\ 1 & 2 \\ 3 & 3 \\ 4 & 1 \\ 3 & 2 \end{bmatrix}
$$

```
      v(p_, i) :=
        Prog
          v_ := [0]
          i := 2
#19:      Loop
            If i > DIM(p_)
              RETURN v_
            v_ := APPEND(v_, [FIRST(REVERSE(v_)) + √((p_↓i↓1 - p_↓(i - 1)↓1)^2 + (p_↓i↓2 - p_↓(i - 1)↓2)^2)])
            i :+ 1
```

#20: SPLINE_PARAM(p_, t_) := VECTOR([(SPLINE([v(p_), VECTOR(p_{i,1}, i, DIMENSION(p_))]', t_)) , (SPLINE([v(p_), VECTOR(p_{i,2}, i, DIMENSION(p_))]'

#21: SPLINE_PARAM(p)

$$
#22: \begin{bmatrix} - 0.02923005489 \cdot t\_^3 \cdot SIGN(t\_ - 2) + 0 \\ 0.03575783648 \cdot t\_^3 \cdot SIGN(416020 \cdot t\_ - 1762289) - 0.3899273482 \cdot t\_^2 \cdot SIGN(416020 \cdot t\_ - 1762289) + 0.8967749163 \cdot t \\ - 0.02279711254 \cdot t\_^3 \cdot SIGN(416020 \cdot t\_ - 1762289) + 0.2252198254 \cdot t\_^2 \cdot SIGN(416020 \cdot t\_ - 1762289) - 0.1990626526 \cdot t\_ \cdot SIGN(416020 \cdot t\_ - 1762289) \\ - 0.05124596352 \cdot t\_^3 \cdot SIGN(365401 \cdot t\_ - 2881680) + 1.212430738 \cdot t\_^2 \cdot SIGN(365401 \cdot t\_ - 2881680) - 9.105607256 \cdot t\_ \cdot SIGN(365401 \cdot t\_ - 288168 \end{bmatrix}
$$

#23: Before parametric plotting display v(p); then take for t_ the intervall 0 < t_ < last element of v(p) (here: 7.88)

#24: v(p)

#25: [0, 2, 4.236067977, 6.472135955, 7.886349517]
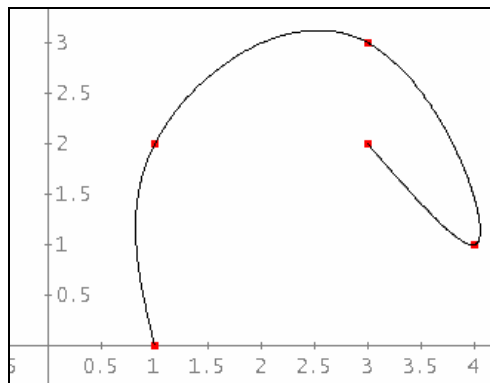
SPLINE(matrix,variable) returns the cubic splines for the points given in matrix with variable variable.

SPLINE_INTERPOLATION(function,variable,list of x-values) produces the cubic spline approximation of function with variable.

SPLINE_PARAM(matrix,variable) returns the parametric splines. The domain of the variable for plotting is from 0 to the last number which appears in v(matrix).

Mr Klingen won the competition. I don´t hesitate to award him the virtual

Golden Spline.

Jan Vermeylen from Belgium created this nice cat using a polygon consisting of 71 points. First I plot the points in discrete and then in connected mode.
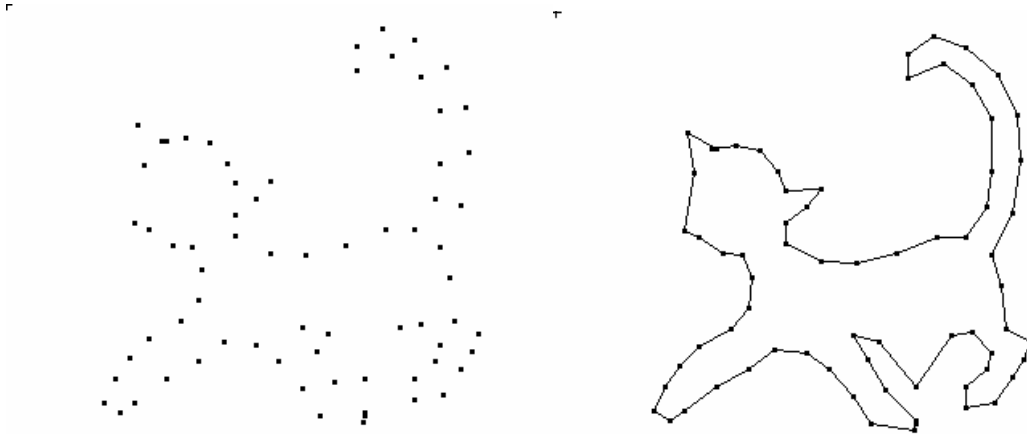
Then I used Leo Klingens PARAMSPLI to give the cat a smoother contour. I couldn´t do it in one step – this is too much for DERIVE´s memory. I had to take the cat to pieces:
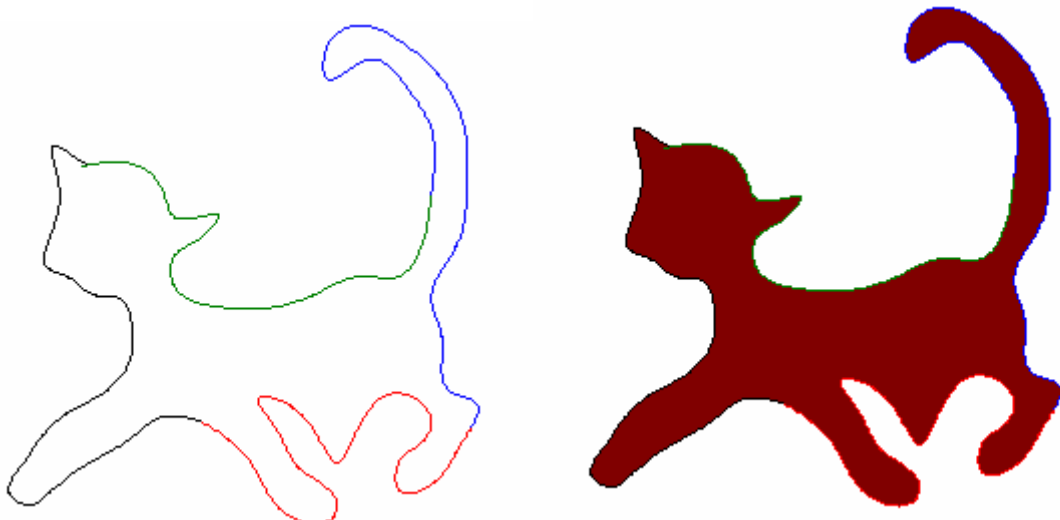
```
        cat1 := cat
#16:              [1, ..., 20]

#17:  FIRST(REVERSE(v(cat1)))

#18:  19.217

#19:  SPLINE_PARAM(cat1, t_)

        cat2 := cat
#20:              [20, ..., 40]

#21:  FIRST(REVERSE(v(cat2)))

#22:  20.248
```

You can distinguish see the four pieces by the colours.

The right cat was filled with colour applying the conversion to a picture object (Edit Menu) followed by little work with MS Paintbrush.

and so on. cat is the matrix of the points.

Ein Zugang zu den Beziérkurven mit DERIVE
An Approach to the Beziér Curves with DERIVE

Günter Scheu, Pfinztal, Germany

1. Einleitung

Mit der Verbreitung der frei skalierbaren True- Type-Schriften wurden auch die Bézierkurven allgemein bekannt, da sie auch zu deren Darstellung benutzt werden. In den sechziger Jahren haben Paul de Casteljau bei Citroën und Pierre Etienne Bézier bei Renault unabhängig voneinander dieses Konzept der Dar- stellung von Kurven und Flächen entwickelt. Sie haben sich als Konstrukteure in der Autoindustrie mit dem Problem beschäftigt, Autos am Zeichenbrett zu konstruieren, Automodelle zu entwerfen und Autoteile mit NC-Werkzeugmaschinen zu produzieren (ähnliche Probleme gab es wohl auch in der Flugzeugindustrie). Diese Produktionsschritte ließen sich dadurch enorm vereinfachen, dass es gelang, die Kurven und Flächen einfach zu beschreiben, zu konstruieren und mit wenigen Punkten abzuspeichern.

Mit der Methode von de Casteljau und Bézier können beliebige Kurven und Flächen mit Hilfe der Bézierpolynome dargestellt werden. Diese Darstellung ist eine Approximation der gegebenen Kurve bzw . Fläche, da diese i.a. nicht analytisch darstellbar sein müssen. Die Polynome tragen den Namen von Bézier, der im Gegensatz zu de Casteljau seine Ergebnisse veröffentlichen konnte (Citroën behielt die Ergebnisse von de Casteljau geheim, sie wurden erst ab 1975 bekannt).

Wir wollen die wichtigsten Eigenschaften der Bézierpolynome aufzeigen und damit Kurven als Funktion eines Parameters entwerfen und zeichnen.

Together with the spreading of the free scaleable True Type fonts the Bézier curves became familiar because they are used for presentation of these fonts. In the sixties Paul de Castejau at Citroën and Pierre Etienne Bézier at Renault developed independently this concept of representing curves and surfaces. As designers and constructors working in car-industry they dealt with the problem to design cars on the drawing board, to construct their models and to produce the parts of the cars using NC-machines. These production steps could be simplified in a high degree because it was now possible to describe curves and surfaces in an easy way and to save them using only a few points.

Applying the method of de Casteljau and Bézier any arbitrary function can be described by Bézier polynomials. This is an approximation of the curves or surfaces wanted which have not to be analytically describable. The polynomials are called Bézier polynomials because Bézier published his results while de Casteljau was forced by Citroën to keep his work secret until 1975.

We want to show the most important properties of the Bézier polynomials and then design and plot curves as functions of one parameter. We use Bernstein polynomials which have only one maximum value in the given interval and are passing the endpoints of the interval.

2. Bèzierpolynome

Bézierpolynome werden mit Hilfe der Bernsteinpolynome dargestellt, die im gegebenen Intervall nur ein Maximum haben und durch die Endpunkte des Intervalls gehen.

Die Bernsteinpolynome lauten mit den Binomialkoeffizienten $\binom{n}{i} = \dfrac{n!}{i!(n-i)!}$:

$$B^n_i(t) = \binom{n}{i}(1-t)^{n-i}\, t^i,\ \ t \in [0,1],\ i = 0,1,\dots,n. \ \text{(\textit{the Bernstein polynomials})}$$

Es sind Polynome vom Grad n mit den Eigenschaften (*Polynomials of degree n with properties*)

a) $\displaystyle \sum_{i=0}^{n} B^n_i(t) = \sum_{i=0}^{n}\binom{n}{i}(1-t)^{n-i}\, t^i = ((1-t)+t)^n = 1$

b) $B^n_i(t)$ hat eine i-fache Nullstelle für $t = 0$, (*a zero of order i for t = 0*)

c) $B^n_i(t)$ hat eine n–i-fache Nullstelle für $t = 1$, (*a zero of order n-1 for t = 1*)

d) $B^n_i(t)$ hat an der Stelle $t = \dfrac{i}{n}$ sein einziges Maximum, (***sole maximum** at $t = \dfrac{i}{n}$*)

e) $B^n_i(t) = B^n_{n-i}(1-t)$, d.h. sie sind symmetrisch bez. t und $t - 1$. (***symmetric wrt t and t – 1***)

Wir veranschaulichen uns grafisch einige Bernsteinpolynome.
Die Beispiele sind in DERIVE-Notation

Visualization of some Bernstein polynomials (in DERIVE notation)

```
#1:   B(i, n, t) := COMB(n, i)·(1 - t)^(n - i)·t^i

#2:   VECTOR(B(i, 4, t), i, 0, 4)

#3:   VECTOR(B(i, 5, t), i, 0, 5)

#4:   F(t, n) := VECTOR([t, B(i, n, t)], i, 0, n)

#5:   F(t, 4)

#6:   F(t, 5)
```
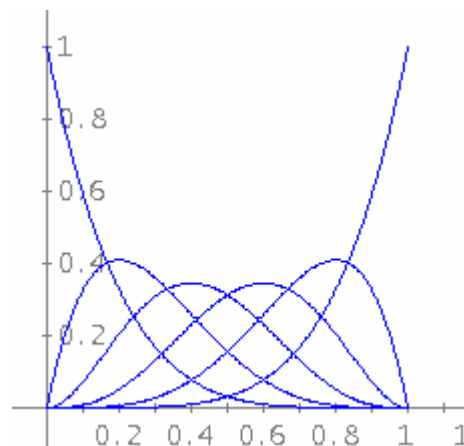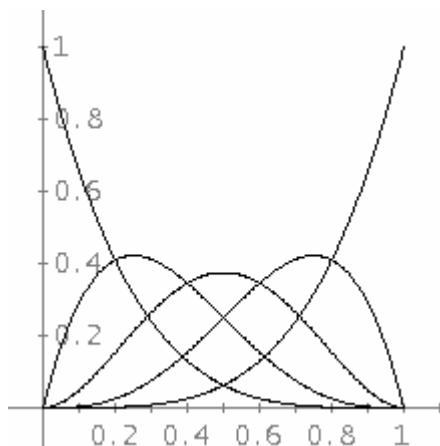

Die 5 Bernsteinpolynome vom Grad 4 und die 6 Bernsteinpolynome vom Grad 5

Die Bernsteinpolynome genügen der Rekursion:

A recursion formula for creating the Bernstein polynomials:

$$B^n_i(t) = (1-t)B^{n-1}_i(t) + t \cdot B^{n-1}_{i-1}(t) \ \text{ mit } \ B^0_0(t) = 1, \ B^n_j(t) = 0 \ \text{ für } j > n, \ t \in [0,1].$$

Die Bernsteinpolynome sind wegen der Eigenschaften b) und c) linear unabhängig.

The Bernstein polynomials are linear independent because of their properties b) and c).

> Das Bézierpolynom vom Grad n ist eine Linearkombination der Bernsteinpolynome vom Grad n.
>
> *The Bézier-polynomial of degree n is a linear combination of Bernstein polynomials of degree n.*
>
> $$p_n(t) = \sum_{i=0}^{n} b_i \cdot B^n_i(t), \ n \in N, \ t \in [0,1]$$

Durch die Bézierkoeffizienten b_i, $i = 0, \ldots, n$ ist das Bézierpolynom und damit die Bézierkurve eindeutig festgelegt.

Die Bézierpolynome haben die Eigenschaften (*Characteristics of the Bézier polynomials*):

A)   In den Endpunkten des Intervalls gelten $p_n(0) = b_0$ und $p_n(1) = b_n$.

B)   Da das Bernsteinpolynom B^n_i bei $t = \dfrac{i}{n}$ sein Maximum hat, sind die Punkte $\left(\dfrac{i}{n}, b_i\right)$,

$i = 0, \ldots, n$ von besonderer Bedeutung. Das Bézierpolynom geht zwar nur durch die Endpunkte $(0, b_0)$ und $(1, b_n)$ aber es approximiert das Bézierpolygon durch die anderen Punkte. Die Punkte des Bézierpolygons liegen äquidistant im Intervall [0,1].

The points (i/n, b_i) are of special importance because they are the maximum points of the Bernstein polynomials. The Bézier polynomial passes only the endpoints (0, b_0) and (1, b_n) but it needs the other points for approximating the Bézier polygon. The points of this Bézier polygon are lying equidistant in [0,1].

C)   Für die Ableitungen in den Endpunkten des Intervalls gelten:
$p'(0) = n(b_1 - b_0)$ und $p'(1) = n(b_n - b_{n-1})$. Die beiden ersten und die beiden letzten Koeffizienten bestimmen die Steigung der Bézierkurve in den Endpunkten des Intervalls. Analoge Gleichungen gelten auch für die höheren Ableitungen.

Both first and last coefficients define the slope of the Bézier curve in the endpoints of the interval. Analogous equations are valid for higher derivatives.

Bézier hat nicht die Bernsteinpolynome as Basis benutzt. Er stellte seine Kurven als Linearkombinationen von Funktionen dar, die die gleiche Rekursion erfüllen wie die Bernsteinpolynome, wobei sich nur die Anfangsbedingungen unterscheiden. R. Forrest entdeckte, dass die Bézier-Form einer Bézierkurve mit Hilfe der Bernsteinpolynome dargestellt werden kann.

Bezier didn´t use Bernstein polynomials as a base. He presented his curves as linear combinations of functions, which meet the same recursion as the Bernstein polynomials but differ in the initial conditions. R. Forrest found out that the Bézier-form of a Bézier curve can be represented by means of Bernstein polynomials.

3. Zeichnen von Kurven mit Bèzierpolynomen

Wir betrachten nur Kurven in der Ebene, d.h. Kurven, die durch zwei Komponenten als Funktion eines Parameters dargestellt werden. Raumkurven werden mit einer Komponente mehr dargestellt.

We will consider curves in the plane only, i.e. curves which can be represented by two components as a function of one parameter. Space curves need one more component.

$$p_n(t) = \sum_{i=0}^{n} b_i B^n_i(t), \ n \in N, t \in [0,1]; \ p_n : [0,1] \to R^2, b_i \in R^2.$$

Wir versuchen die Sinusfunktion im Intervall [0, 2π] durch Bézierkurven zu approximieren.

We try to approximate the sine function in [0, 2π] *by Bézier curves.*

p ist die verwendete Punktematrix. (*p ist he matrix of the given points*).

```
#9:     FB(p, t) := VECTOR(p ·B(i - 1, DIMENSION(p) - 1, t), i, DIMENSION(p))
                       i

                 DIMENSION(p)
#10:  GB(p, t) :=      Σ        (FB(p, t))
                      i=1                  i
```

Die Funktion GB erzeugt die Bézierkurve für *t* ∈ [0,1]. Wir wählen zuerst die Punktmatrix *s1* – für sin(*x*) - und dann *s2* – für 2 sin(*x*). Mit GB(s1) bzw GB(s2) erhalten wir die entsprechenden Bézierkurven.

```
#9:    SIN(x)

#10:   s1 := TABLE(SIN(x), x, 0, 2·π, π/2)

                    ┌                                        ┐
                    │          0                  0          │
                    │                  3                  3  │
                    │  2·π·t·(1 - t)        4·t·(1 - t)      │
#11:  FB(s1) =      │          2          2                  │
                    │  6·π·t ·(t - 1)             0          │
                    │          3                  3          │
                    │  6·π·t ·(1 - t)      4·t ·(t - 1)      │
                    │              4                         │
                    │        2·π·t                0          │
                    └                                        ┘

#12:  GB(s1) = [2·π·t, 4·t·(t - 1)·(2·t - 1)]

#13:   s2 := TABLE(2·SIN(x), x, 0, 2·π, π/2)

#14:  GB(s2) = [2·π·t, 8·t·(t - 1)·(2·t - 1)]
```
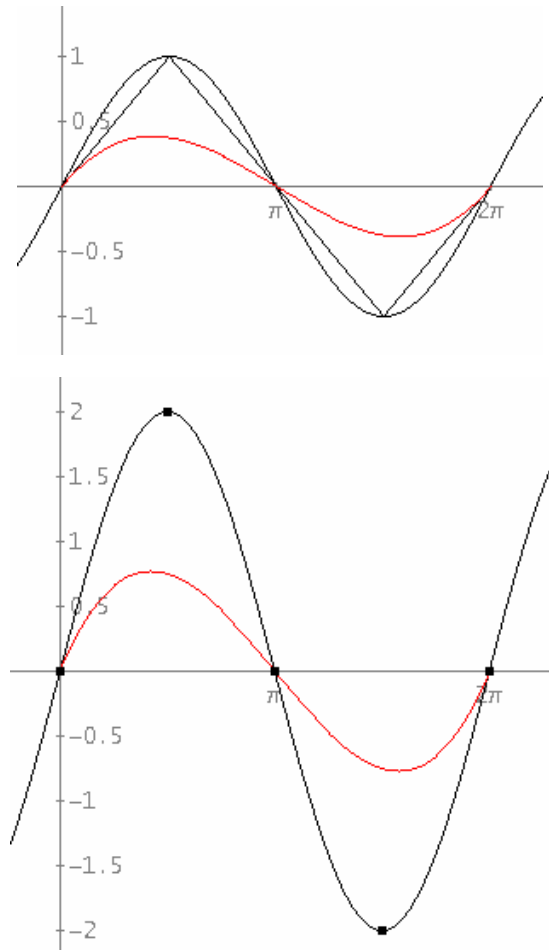
GB gives the Bézier curve for 0 ≤ t ≤ 1t. We choose point matrices s1 and s2 and obtain the two curves calling functions GB(s1) and GB(s2).

Nun wollen wir versuchen, die Konturen von zB Vasen Buchstaben oder Ziffern mit Hilfe zweier Bézierkurven zu erreichen. Dazu wählen wir die beiden Punktmatrizen $q1$ und $q2$ und zeichnen die Kurven wieder mit $0 \le t \le 1$.

Now we will design the contours of eg vases, letters or numbers by two Bézier curves,
(q1 and q3 make a "3") and plot with 0 ≤ t ≤ 1.

$$\#15: \quad q1 := \begin{bmatrix} 1 & 0 \\ 4 & 2 \\ 4 & 6 \\ 1 & 8 \end{bmatrix}, \; q2 := \begin{bmatrix} 1 & 8 \\ 4 & 10 \\ 1 & 10 \end{bmatrix}$$

$$\#16: \quad [GB(q1, x), GB(q2, x)] = \begin{bmatrix} -9 \cdot x^2 + 9 \cdot x + 1 & -2 \cdot x \cdot (2 \cdot x^2 - 3 \cdot x - 3) \\ -6 \cdot x^2 + 6 \cdot x + 1 & -2 \cdot (x^2 - 2 \cdot x - 4) \end{bmatrix}$$
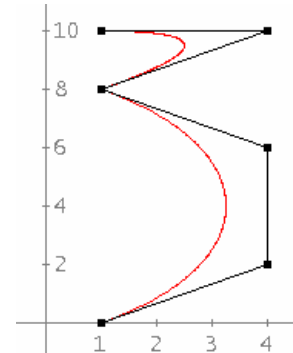
(Wenn man die Punkte unterschiedlich "gewichtet" (= mehrfach in die Matrix aufnimmt) ändert sich der Kurvenverlauf deutlich. Josef)

(If one gives the points different "weights" (= by taking them several times into the matrix) the run of the curve changes significantly. Josef)

$$\#17: \quad GB(APPEND(q1, q2), x)$$

$$\#18: \quad GB\left(\begin{bmatrix} 1 & 0 \\ 4 & 2 \\ 4 & 6 \\ 1 & 8 \\ 4 & 10 \\ 1 & 10 \end{bmatrix}, x\right)$$

$$\#19: \quad GB\left(\begin{bmatrix} 1 & 0 \\ 4 & 2 \\ 4 & 6 \\ 1 & 8 \\ 1 & 8 \\ 1 & 8 \\ 4 & 10 \\ 1 & 10 \end{bmatrix}, x\right)$$

$$\#20: \quad GB\left(\begin{bmatrix} 1 & 0 \\ 4 & 2 \\ 4 & 6 \\ 1 & 8 \\ 1 & 8 \\ 1 & 8 \\ 1 & 8 \\ 1 & 8 \\ 4 & 10 \\ 4 & 10 \\ 1 & 10 \end{bmatrix}, x\right)$$

$$\#21: \quad \left[210 \cdot x^{10} - 1050 \cdot x^9 + 2835 \cdot x^8 - 5040 \cdot x^7 + 5670 \cdot x^6 - 3780 \cdot x^5 + 1260 \cdot x^4 - 135 \cdot x^2 + \right.$$
$$30 \cdot x + 1, \; 2 \cdot x \cdot (17 \cdot x^9 + 50 \cdot x^8 - 315 \cdot x^7 + 480 \cdot x^6 - 210 \cdot x^5 - 252 \cdot x^4 + 420 \cdot x^3 -$$
$$\left. 240 \cdot x^2 + 45 \cdot x + 10) \right]$$

#17   #18   #19   #20
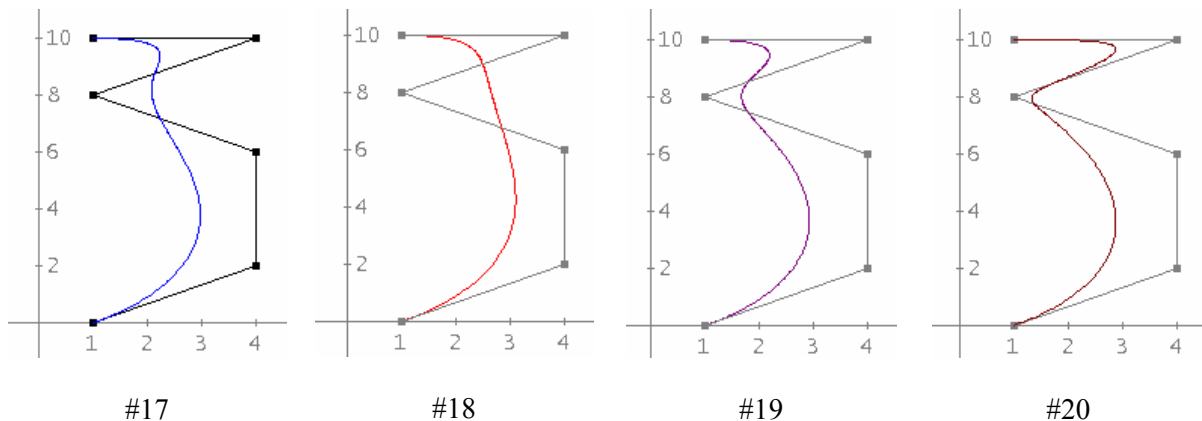
4. Der Algorithmus von Casteljeau

Mit diesem Algorithmus begannen die Arbeiten von de Casteljeau im Jahre 1959. Wir betrachten eine einfache Konstruktion zur Erzeugung einer Parabel. Die Verallgemeinerung dieser Konstruktion führt zu den Bézierkurven und deren numerischer Berechnung.

Es seien b_0, b_1 und b_2 drei beliebige Punkte des Anschauungsraums, die nicht auf einer gemeinsamen Geraden liegen.

This algorithm was the start of de Casteljeau´s work in 1959. We look at an easy construction of a parabola and its generalization will lead us to the construction of Bézier curves and their calculation. Let b_0, b_1 and b_2 three points which are not lying on one common line. We set up the first two equations and substitute for $b_{0,1}$ and $b_{1,2}$ in the third one in order to obtain a parabola as a Bézier curve.

This construction is a repeated linear interpolation. $0 \leq t \leq 1$ gives the parabola within the triangle formed by b_0, b_1 and b_2.

$$b_{0,1}(t) = (1-t)b_0 + t\,b_1$$
$$b_{1,2}(t) = (1-t)b_1 + t\,b_2 \qquad t \in R$$
$$b_{0,1,2}(t) = (1-t)b_{0,1} + t\,b_{1,2}$$

$$b_{0,1,2}(t) = (1-t)^2 b_0 + 2t(1-t)b_1 + t^2 b_2, \quad t \in R.$$

Wir setzen die beiden ersten Gleichungen in die dritte ein und erhalten mit dem quadratischen Term eine Parabel.

Diese Konstruktion ist eine wiederholte lineare Interpolation. Wenn der Parameter t zwischen 0 und 1 läuft, dann liegt die zugehörige Parabel innerhalb des von den drei Punkten b_0, b_1 und b_2 gebildeten Dreiecks. Dabei gilt $b_{0,1,2}(t=0) = b_0$ und $b_{0,1,2}(t=1) = b_2$.

Allgemein seien mit b_0, b_1, …, b_n insgesamt $n+1$ Punkte der Ebene gegeben. Wir bilden damit den Ausdruck

$$b_{0,\ldots,i}(t) = (1-t)b_{0,\ldots,i-1}(t) + t\,b_{1,\ldots,i}, \quad 1 \leq i \leq n.$$

Dann ist $b_{0,1,\ldots,n}(t)$ der Punkt der Bézierkurve zum Parameterwert $t \in [0,1]$. Das Polygon durch die Bézierpunkte b_0, b_1, …, b_n i st das Bézierpolygon. Die Punkte der Bézierkurve zum Parameterwert t werden mit dem **de Casteljeau Schema** berechnet.

$$1-t \begin{array}{|llll} b_0 & & & \\ b_1 & b_{0,1} & & \\ b_2 & b_{1,2} & b_{0,1,2} & \\ b_3 & b_{2,3} & b_{1,2,3} & b_{0,1,2,3} \\ b_4 & b_{3,4} & b_{2,3,4} & b_{1,2,3,4} \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ b_n & b_{n-1,n} & b_{n-2,n-1,n} & ... \end{array}$$

$$\rightarrow b_{0,1,...,n} = p_n(t),\ t \in [0,1]$$

Bei der numerischen Berechnung wird nicht die vollständige Dreiecksmatrix benötigt, denn die erste Spalte kann bei jedem Schritt geeignet überschrieben werden. Es muss daher keine Tabelle abgespeichert werden.

(Zur Illustration kann der Algorithmus mit Schülern händisch für vier Punkte $b_0 = (1,1)$, $b_1 = (1.5,0)$, $b_2 = (3,1)$ und $b_4 = (4,2.5)$ und $t = 0,4$ durchgeführt werden. Das ließe sich natürlich auch für den allgemeinen Parameter t durchführen. Josef)

Given are n + 1 points b_0, b_1, …, b_n. They form the Bézier polygon. The Casteljeau table returns the points of the Bézier curve for parameter $t \in [1,0]$. This algorithm is more efficient. It is based upon a repeated linear interpolation. Neither equidistant nodes nor Bézier polynomials are needed.

(As an illustration I´d recommend to perform this algorithm with pupils manually for eg four points: $b_0 = (1,1)$, $b_1 = (1.5,0)$, $b_2 = (3,1)$ and $b_4 = (4,2.5)$ and $t = 0.4$. One could – should? – do this also producing the iterated curves. Josef)

$t = 0.4$ $\quad \begin{pmatrix}1\\1\end{pmatrix}$

$\begin{pmatrix}1.5\\0\end{pmatrix} \quad 0.6 \cdot \begin{pmatrix}1\\1\end{pmatrix} + 0.4 \cdot \begin{pmatrix}1.5\\0\end{pmatrix} = \begin{pmatrix}1.2\\0.6\end{pmatrix}$

$1-t = 0.6 \quad \begin{pmatrix}3\\1\end{pmatrix} \quad 0.6 \cdot \begin{pmatrix}1.5\\0\end{pmatrix} + 0.4 \cdot \begin{pmatrix}3\\1\end{pmatrix} = \begin{pmatrix}2.1\\0.4\end{pmatrix} \quad 0.6 \cdot \begin{pmatrix}1.2\\0.6\end{pmatrix} + 0.4 \cdot \begin{pmatrix}2.1\\0.4\end{pmatrix} = \begin{pmatrix}1.56\\0.52\end{pmatrix}$

$\begin{pmatrix}4\\2.5\end{pmatrix} \quad 0.6 \cdot \begin{pmatrix}3\\1\end{pmatrix} + 0.4 \cdot \begin{pmatrix}4\\2.5\end{pmatrix} = \begin{pmatrix}3.4\\1.6\end{pmatrix} \quad 0.6 \cdot \begin{pmatrix}2.1\\0.4\end{pmatrix} + 0.4 \cdot \begin{pmatrix}3.4\\1.6\end{pmatrix} = \begin{pmatrix}2.62\\0.88\end{pmatrix} \quad 0.6 \cdot \begin{pmatrix}1.56\\0.52\end{pmatrix} + 0.4 \cdot \begin{pmatrix}2.62\\0.88\end{pmatrix} = \begin{pmatrix}1.984\\0.664\end{pmatrix}$

$\begin{pmatrix}1\\1\end{pmatrix}$

$\begin{pmatrix}1.5\\0\end{pmatrix} \quad (1-t)\begin{pmatrix}1\\1\end{pmatrix} + t\begin{pmatrix}1.5\\0\end{pmatrix} = \begin{pmatrix}0.5t+1\\-t+1\end{pmatrix}$

$\begin{pmatrix}3\\1\end{pmatrix} \quad (1-t)\begin{pmatrix}1.5\\0\end{pmatrix} + t\begin{pmatrix}3\\1\end{pmatrix} = \begin{pmatrix}1.5t+1.5\\t\end{pmatrix} \quad (1-t)\begin{pmatrix}0.5t+1\\-t+1\end{pmatrix} + t\begin{pmatrix}1.5t+1.5\\t\end{pmatrix} = \\ = \begin{pmatrix}t^2+t+1\\2t^2-2t+1\end{pmatrix}$

$\begin{pmatrix}4\\2.5\end{pmatrix} \quad (1-t)\begin{pmatrix}3\\1\end{pmatrix} + t\begin{pmatrix}4\\2.5\end{pmatrix} = \begin{pmatrix}t+3\\1.5t+1\end{pmatrix} \quad (1-t)\begin{pmatrix}1.5t+1.5\\t\end{pmatrix} + t\begin{pmatrix}t+3\\1.5t+1\end{pmatrix} = \\ = \begin{pmatrix}-0.5t^2+3t+1.5\\0.5t^2+2t\end{pmatrix} \quad (1-t)\begin{pmatrix}t^2+t+1\\2t^2-2t+1\end{pmatrix} + t\begin{pmatrix}-0.5t^2+3t+1.5\\0.5t^2+2t\end{pmatrix} = \\ = \begin{pmatrix}-1.5t^3+3t^2+1.5t+1\\-1.5t^3+6t^2-3t+1\end{pmatrix}$

Wir definieren die Funktion CB(p,t) – sie liefert den Wert – und RB(p,t,a) – sie liefert die *a*-te Spalte der Tabelle für die Punktmatrix *p*.

We define function CB(p,t) – returning the value – and RB(p,t,a) – returning the ath column of the table for given point matrix p.

#23: $CB(p, t) := ITERATE(VECTOR((1 - t) \cdot x_j + t \cdot x_{j+1}, j, DIM(x) - 1), x, p, DIM(p) - 1)$

#24: $RB(p, t, a) := ITERATE(VECTOR((1 - t) \cdot x_j + t \cdot x_{j+1}, j, DIM(x) - 1), x, p, a)$

#26: $p1 := \begin{bmatrix} 1 & 1 \\ 1.5 & 0 \\ 4 & 1 \end{bmatrix}$

#27: [RB(p1, 0.6, 0), RB(p1, 0.6, 1), RB(p1, 0.6, 2)]

#28: $\left[\begin{bmatrix} 1 & 1 \\ \dfrac{3}{2} & 0 \\ 4 & 1 \end{bmatrix}, \begin{bmatrix} \dfrac{13}{10} & \dfrac{2}{5} \\ 3 & \dfrac{3}{5} \end{bmatrix}, \left[\begin{bmatrix} \dfrac{58}{25} & \dfrac{13}{25} \end{bmatrix} \right] \right]$
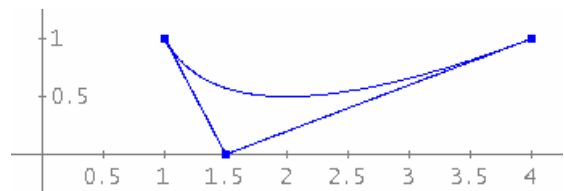
#29: [RB(p1, t, 0), RB(p1, t, 1), RB(p1, t, 2)]

#30: $\left[\begin{bmatrix} 1 & 1 \\ \dfrac{3}{2} & 0 \\ 4 & 1 \end{bmatrix}, \begin{bmatrix} \dfrac{t}{2} + 1 & 1 - t \\ \dfrac{5 \cdot t + 3}{2} & t \end{bmatrix}, \left[\begin{bmatrix} 2 \cdot t^2 + t + 1 & 2 \cdot t^2 - 2 \cdot t + 1 \end{bmatrix} \right] \right]$

#31: $CB(p1) = \begin{bmatrix} 2 \cdot t^2 + t + 1 & 2 \cdot t^2 - 2 \cdot t + 1 \end{bmatrix}$
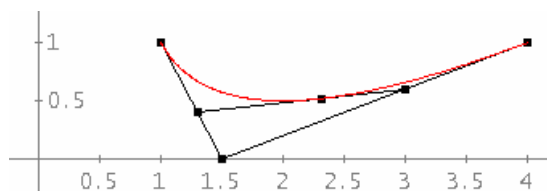
#28 & #31   #30

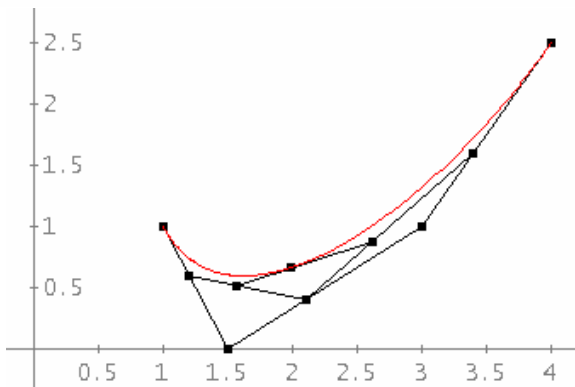#32: $p2 := \begin{bmatrix} 1 & 1 \\ 1.5 & 0 \\ 3 & 1 \\ 4 & 2.5 \end{bmatrix}$

#33: [RB(p2, 0.4, 0), RB(p2, 0.4, 1), RB(p2, 0.4, 2), RB(p2, 0.4, 3)]

#34: $\left[\begin{bmatrix} 1 & 1 \\ 1.5 & 0 \\ 3 & 1 \\ 4 & 2.5 \end{bmatrix}, \begin{bmatrix} 1.2 & 0.6 \\ 2.1 & 0.4 \\ 3.4 & 1.6 \end{bmatrix}, \begin{bmatrix} 1.56 & 0.52 \\ 2.62 & 0.88 \end{bmatrix}, [[1.984, 0.664]] \right]$
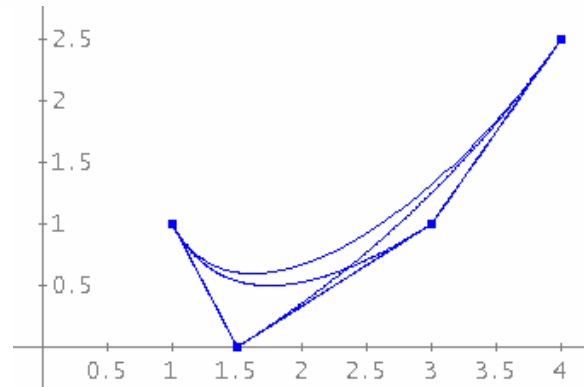
#35: $CB(p2, t) = \left[-\dfrac{3 \cdot t^3 - 6 \cdot t^2 - 3 \cdot t - 2}{2}, -\dfrac{3 \cdot t^3 - 12 \cdot t^2 + 6 \cdot t - 2}{2} \right]$

#36:   [RB(p2, t, 0), RB(p2, t, 1), RB(p2, t, 2), RB(p2, t, 3)]

#37:
$$\left[\begin{bmatrix} 1 & 1 \\ 1.5 & 0 \\ 3 & 1 \\ 4 & 2.5 \end{bmatrix}, \begin{bmatrix} 0.5 \cdot t + 1 & 1 - t \\ 1.5 \cdot (t + 1) & t \\ t + 3 & 1.5 \cdot t + 1 \end{bmatrix}, \begin{bmatrix} t^2 + t + 1 & 2 \cdot t^2 - 2 \cdot t + 1 \\ -0.5 \cdot (t^2 - 6 \cdot t - 3) & 0.5 \cdot t \cdot (t + 4) \end{bmatrix}, \right.$$

$$\left. \begin{bmatrix} -0.5 \cdot (3 \cdot t^3 - 6 \cdot t^2 - 3 \cdot t - 2), & -0.5 \cdot (3 \cdot t^3 - 12 \cdot t^2 + 6 \cdot t - 2) \end{bmatrix} \right]$$

#34 & #35   #37

Vergleiche mit den oben gefundenen händischen Ergebnissen – Compare with the results above derived manually

5. Gradanhebung – *Raising the degree*

Wir fügen einen Punkt zum Bézierpolynom hinzu, ohne die Bézierkurve zu ändern, wobei wir aber den Grad um 1 erhöhen. Das ergibt neue Punkte b_i^1, die der Gleichung (1) genügen.

(1) $\qquad \sum_{i=0}^{n} b_i \binom{n}{i} (1-t)^{n-i} t^i = \sum_{i=0}^{n+1} b_i^1 \binom{n+1}{i} (1-t)^{n+1-i} t^i.$

Wir multiplizieren die linke Seite mit $t + (1 - t)$ und erhalten durch Koeffizientenvergleich die Gleichungen (2) und (3).

(2) $\qquad \binom{n+1}{i} b_i^1 = \binom{n}{i} b_i + \binom{n}{i-1} b_{i-1} \qquad$ oder – *or*

(3) $\qquad b_i^1 = \dfrac{1}{n+1} b_i + \left(1 - \dfrac{1}{n+1}\right) b_{i-1}, \ _{i=0,...,n+1}$

We add one point to the Bézier polygon without changing the Bézier curve. Doing this we raise the degree by 1. We find new points b_i^1 which meet equation (1). Multiplying its left side by t + (1 – t) and comparing the coefficients leads to (2) and (3). The new points are emerging from the old ones by linear interpolation at the values i/(n+1). Raising the degree is important for numerical calculations because many programs require all Bézier polygons having the same degree (for exchangig data purposes).

Dann veranschaulichen wir an einem Beispiel die Polygone, die bei der Gradanhebung entstehen. Gegeben sei die Punktmatrix $p3$. Als neue Funktion wird MP(p) definiert, die die Eckpunkte des neuen Polygons liefert. Dann zeichnen wir die zugehörige Parabel bzw. den Parabelteil und die Polygone, die bei der Gradanhebung entstehen. Diesen Vorgang können wir beliebig oft wiederholen

MP(p) returns the vertices of the new polygon. Then we plot the parabola and the new polygon. This procedure can be repeated arbitrarily often.

#38: $MB(p) := VECTOR\left(IF\left(j = 0, 0, \dfrac{j}{DIM(p)} \cdot p_j\right) + \left(1 - \dfrac{j}{DIM(p)}\right) \cdot p_{j+1}, j, 0, DIM(p)\right)$

#39: $p3 := \begin{bmatrix} 0 & 0 \\ 2 & 1 \\ 0 & 4 \end{bmatrix}$

#40: $GB(p3, t) = [4 \cdot t \cdot (1 - t), 2 \cdot t \cdot (t + 1)]$

#41: $(p4 := MB(p3)) = p4 := \begin{bmatrix} 0 & 0 \\ \dfrac{4}{3} & \dfrac{2}{3} \\ \dfrac{4}{3} & 2 \\ 0 & 4 \end{bmatrix}$

#42: $GB(p4, t) = [4 \cdot t \cdot (1 - t), 2 \cdot t \cdot (t + 1)]$

#43: $(p5 := MB(p4)) = p5 := \begin{bmatrix} 0 & 0 \\ 1 & \dfrac{1}{2} \\ \dfrac{4}{3} & \dfrac{4}{3} \\ 1 & \dfrac{5}{2} \\ 0 & 4 \end{bmatrix}$

#44: $(p6 := MB(p5)) = p6 := \begin{bmatrix} 0 & 0 \\ \dfrac{4}{5} & \dfrac{2}{5} \\ \dfrac{6}{5} & 1 \\ \dfrac{6}{5} & \dfrac{9}{5} \\ \dfrac{4}{5} & \dfrac{14}{5} \\ 0 & 4 \end{bmatrix}$

Wir fahren fort bis zum Polygon $p10$, berechnen wieder GB(p10) und stellen fest, dass sich immer wieder die gleiche Parabel als Bézierkurve ergibt. Alles gemeinsam kann in unterschiedlichen Fraben gezeichnet werden.

We proceed until polygon p10, calculate again GB(p10) and observe that the result is the same parabola as Bézier curve. Finally we plot all together using different colours.
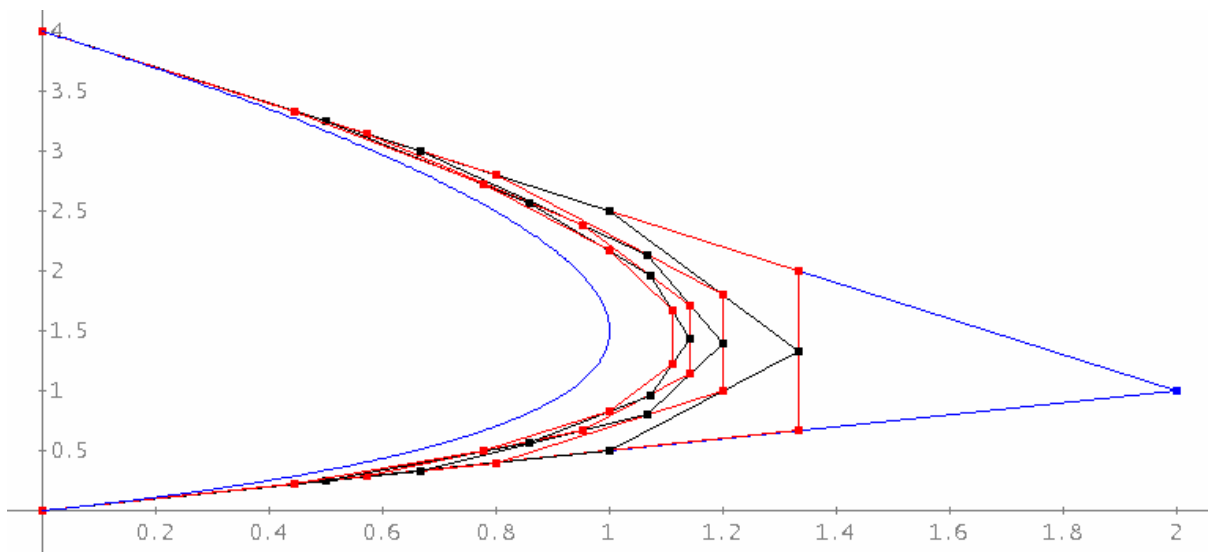
Alle Polygone, die jeweils durch Gradanhebung auseinander hervorgehen gemeinsam mit der Bézier-kurve – hier eine Parabel.

6. Ausblicke – *Prospects*

Es ist einfach, die Bézierkurven auf beliebige Intervalle [*a*,*b*] zu verallgemeinern. Ableitungen und Integrale von Bézierkurven können noch untersucht werden. Werden mehr als 10 Punkte zur Konstruktion einer Bézierkurve verwendet, dann werden die Berechnungen zu aufwändig und es werden Splinekurven in Bézier-Form benützt. Dies und weitere Untersuchungen sind in dem Buch von Farin teilweise durchgeführt.

It is easy to generalize Bézier curves for arbitrary intervals [a,b]. Derivatives and integrals can be investigated. If one uses more than 10 points for designing a curve then the calculations will become too bulky and one has to use splines in Bézier form instead. These and other investigations can be found in Farin´s book.

7. Literaturverzeichnis – *References*

Farin G., *Kurven und Flächen im Computer Aided Geometric Design. Eine praktische Einführung*, Vieweg 1994

Kose K., Schröder R., Wieiczek K., *Numerik sehen und verstehen. Ein kombiniertes Lehr- und Arbeitsbuch*, Vieweg 1992

More books (also treating splines):

Luther W., Ohlsmann M., *Mathematische Grundlagen der Computergraphik*, Vieweg 1989

Müller B. (editor), *Techniken der Computergrafik*, Franzis-Verlag 1994

Schonefeld S., *Numerical Analysis via DERIVE*, Math Ware 1994 (only Splines)

Hill F., *Computer Graphics*, Macmillan Publishing Company 1990

Weber M., *3-D-Grafik Theorie und Praxis*, IWT 1985

Recommended websites:

`http://www.geometry.at/ahsktn/bezier/praesentation_copyright.pdf`

`http://www.dma.ufg.ac.at/app/link/Grundlagen:2D-Grafik/module/13079?step=3`

In times of DERIVE 6.10 we can additionally introduce slider bars to visualize how the form of the Bézier polygon influences the form of the Bézier curve. You can compare with working with a graphics program. The origin is a fixed point of the polygon, the other three points can be varied by using the slider bars. Josef

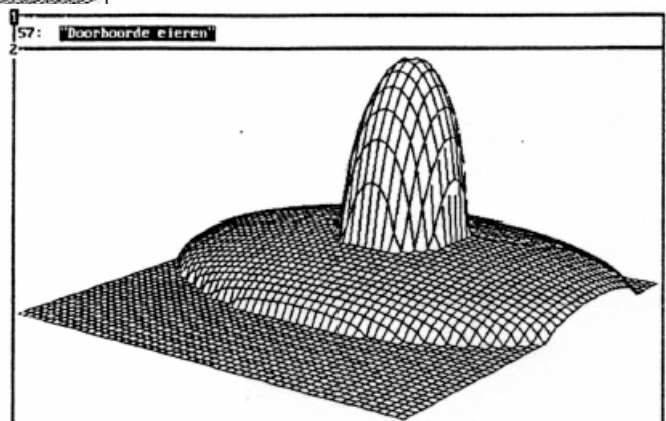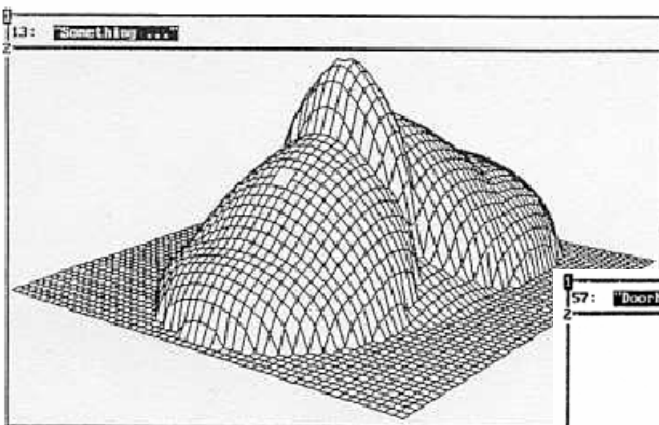**Working with slider bars for the coordinates:**

#50: [x1 :=, y1 :=, x2 :=, y2 :=, x3 :=, y3 :=]

$$
\#51: \quad \text{pts} := \begin{bmatrix} 0 & 0 \\ x1 & y1 \\ x2 & y2 \\ x3 & y3 \end{bmatrix}
$$

#52: GB(pts)
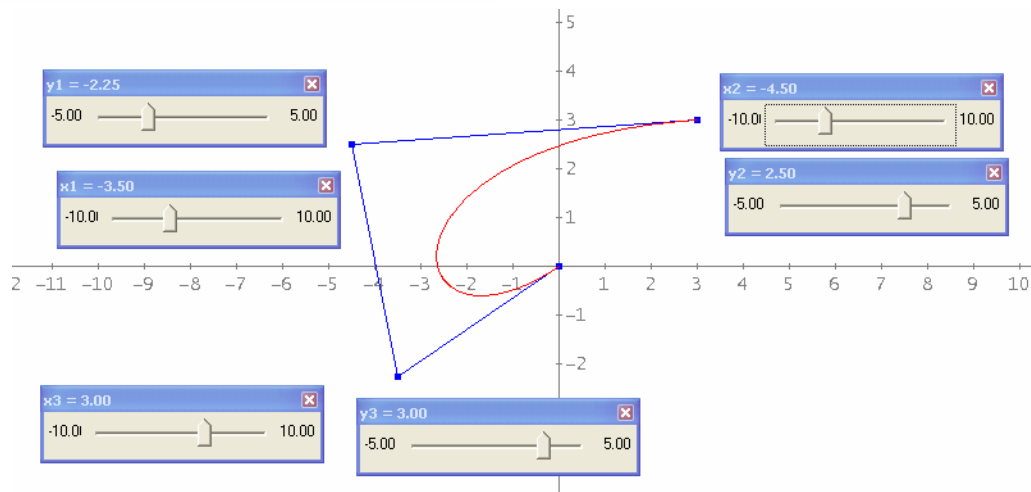
See also DNLs # 53 and #66.

Anmerkungen zu Thomas Weths Kurven aus DNL#14
Comments on Thomas Weth´s curves in DNL#14

Peter Baum, Kassel, Germany

Peter Baum sent a contribution dealing with Thomas Weths´article in the DERIVE Newsletter #14. It is always a pleasure for the editor so see that the DNL is a source for the readers´ own investigations. Peter sent the MTH-file with German comments and many TIF-files of the plots. I try to translate his comments connecting the DERIVE commands for our many non German speaking friends. Josef.

The mapping described by Thomas Weth in DNL#14 (inversion on a circle followed by an reflection wrt the origin), can easily be represented by means of polar coordinates. Even the mapping of circles is very clear. I restrict myself to the description, the reader might try to derive the formulae. I will start with the mapping of straight lines.
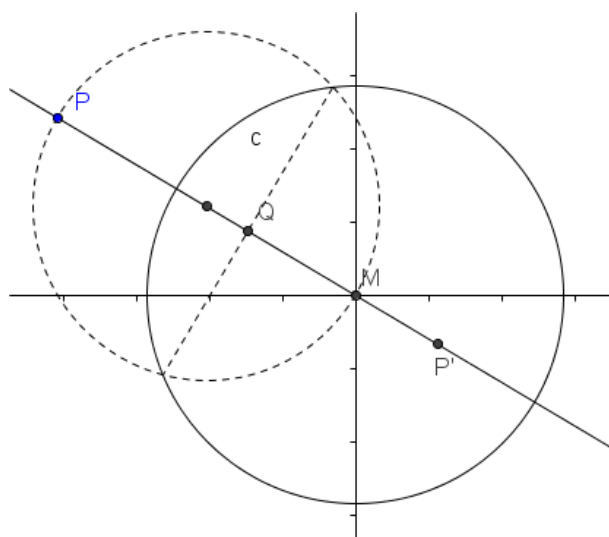
Auxiliary point *Q* is obtained by inversion of *P* wrt to the circle with centre in the origin.
The final mapping of *P* is *P'* as result of a reflection with centre M.

Let (s, α) the pre-image (= german: Urbild) of a point and (r, α) its image. The circle of inversion with radius $R = a$ has its centre in the origin. Then we obtain the equation of the mapping described above

#1: $$r = \frac{2 \cdot a^2}{s} - s$$

If the picture of a point lies in the origin (pole) then $r = 0$, which leads to $s = a\sqrt{2}$. All these points form a circle concentric to the inversion circle with radius $s = a\sqrt{2}$. I will call this circle *pole circle*.

#2: $r = a$

#3: $r = a \cdot \sqrt{2}$

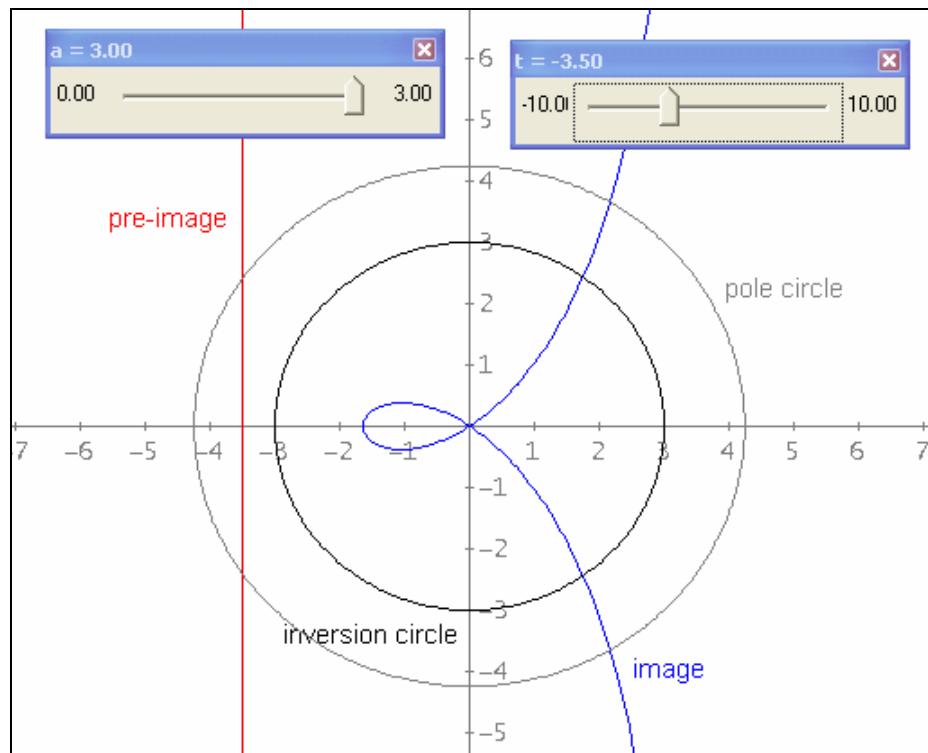#2 and #3 give these two circles (in polar coordinates, of course).

1. **The pre-image curve is a straight line parallel to $y = 0$ with distance t.**

Its equation in polar coordinates is given by #4 and simplifying #1 returns the equation of the image curve:

#4: $\quad s := \dfrac{t}{\cos(\alpha)}$

#5: $\quad r = \dfrac{2 \cdot a^2 \cdot \cos(\alpha)}{t} - \dfrac{t}{\cos(\alpha)}$

Thanks the slider bars we can now easily animate this mapping.

The picture has a vertex if the line is a tangent of the pole circle. Then $t = a\sqrt{2}$. By substituting for $t = a\sqrt{2}$ in #5 we obtain the equation of a kissoid.

#6: $\quad r = \sqrt{2} \cdot a \cdot \cos(\alpha) - \dfrac{\sqrt{2} \cdot a}{\cos(\alpha)}$

#7: $\quad r = -\sqrt{2} \cdot a \cdot \sin(\alpha) \cdot \tan(\alpha)$

#8: $\quad$ Trigpower := Cosines

#9: $\quad r = \sqrt{2} \cdot a \cdot \cos(\alpha) - \dfrac{\sqrt{2} \cdot a}{\cos(\alpha)}$

Equation #7 is the well known form of a kissoid and by simplifying it – forcing the cosine – we can observe the identity of both forms.

Substituting $t = a$ in #5 leads to a strophoid:

```
#10:   r = 2·a·COS(α) -  ──────
                         COS(α)
                           a
```

```
#11:   Trigonometry := Collect
```

```
#12:   r = 2·a·COS(α) -  ──────
                         COS(α)
                           a
```

Former DERIVE versions simplified under Trig Options Collect to the form $r = \dfrac{a\cos(2\alpha)}{\cos(\alpha)}$.

DERIVE 6 does not. So we need another way to show the identity:

```
#11:   Trigonometry := Collect
```

```
#12:   r = 2·a·COS(α) -  ──────
                         COS(α)
                           a
```

```
#13:   ⎛2·a·COS(α) -  a  ⎞  -  a·COS(2·α)  =  -  a·COS(2·α)  + 2·a·COS(α) -   a
       ⎝            COS(α)⎠     COS(α)           COS(α)                     COS(α)
```

```
#14:   Trigonometry := Expand
```

```
#15:   0 = 0
```

Substituting $t = \dfrac{a}{\sqrt{2}}$ in #5 leads us to the equation of the trisectrix:

```
                                  √2·a
#16:   r = 2·√2·a·COS(α) -  ──────────
                           2·COS(α)
```

2.   **Now the pre-image curve is a circle with its centre in M(*t*,0) and radius *b*.**

The equation of the circle in polar coordinates is given by expression #6:

```
#1:    LOAD(D:\DfD\DNL\DNL95\MTH19\DOT_CIRC.MTH)
```

```
                2
              2·a
#2:    r =   ──────  - s
               s
```

```
#3:    r = a
```

```
#4:    KSTRX(0, 0, a·√2, 30, 0.6)
```

```
#5:    r = a·√2
```

```
                             2    2      2
#6:    s := t·COS(α) + √(b  - t ·SIN(α) )
```

#3 and #5 are inversion circle and pole circle. #4 is a function which plots the circle as
a dashed line. This function is part of a little toolbox which will be described at the end
of this contribution. Josef

Simplifying expression #2 using the appropriate Mode Settings will lead us to the general equation of the picture curve:

#7:   Trigonometry := Auto

$$\#8: \quad r = -\frac{t^2 \cdot \cos(\alpha)^2 + t \cdot \cos(\alpha) \cdot \sqrt{(b^2 - t^2 \cdot \sin(\alpha)^2)} - 2 \cdot a^2}{t \cdot \cos(\alpha) + \sqrt{(b^2 - t^2 \cdot \sin(\alpha)^2)}} - \sqrt{(b^2 - t^2 \cdot \sin(\alpha)^2)}$$

$$\#9: \quad r = -\frac{2 \cdot t^2 \cdot \cos(\alpha)^2 + 2 \cdot t \cdot \cos(\alpha) \cdot \sqrt{(b^2 - t^2 \cdot \sin(\alpha)^2)} - 2 \cdot a^2 + b^2 - t^2}{t \cdot \cos(\alpha) + \sqrt{(b^2 - t^2 \cdot \sin(\alpha)^2)}}$$

#10:   Trigonometry := Collect

$$\#11: \quad r = -\frac{\sqrt{2} \cdot (\sqrt{2} \cdot t \cdot \cos(\alpha) \cdot \sqrt{(t^2 \cdot \cos(2 \cdot \alpha) + 2 \cdot b^2 - t^2)} + t^2 \cdot \cos(2 \cdot \alpha) - 2 \cdot a^2 + b^2)}{\sqrt{(t^2 \cdot \cos(2 \cdot \alpha) + 2 \cdot b^2 - t^2)} + \sqrt{2} \cdot t \cdot \cos(\alpha)}$$

$$\#12: \quad r = -\frac{\sqrt{2} \cdot (\sqrt{2} \cdot t \cdot \cos(\alpha) \cdot \sqrt{(t^2 \cdot (2 \cdot \cos(\alpha)^2 - 1) + 2 \cdot b^2 - t^2)} + t^2 \cdot (2 \cdot \cos(\alpha)^2 - 1) - 2 \cdot a^2 + b^2)}{\sqrt{(t^2 \cdot (2 \cdot \cos(\alpha)^2 - 1) + 2 \cdot b^2 - t^2)} + \sqrt{2} \cdot t \cdot \cos(\alpha)}$$

#13:   Trigonometry := Expand

#14:   Trigpower := Cosines

$$\#15: \quad r = \frac{(2 \cdot a^2 - b^2 + t^2) \cdot \sqrt{(t^2 \cdot \cos(\alpha)^2 + b^2 - t^2)} - t \cdot (2 \cdot a^2 + b^2 - t^2) \cdot \cos(\alpha)}{b^2 - t^2}$$

Derive for DOS performed a little bit different because the identity $\cos(2\alpha) = 2\cos^2(\alpha) - 1$ was considered in both directions. The next expressions are from the original 1995 file.

#16:   Trigonometry := Collect

#17:   and Factor Trivial

$$\#18: \quad r = \frac{\sqrt{2} \cdot ((2 \cdot a^2 - b^2 + t^2) \cdot \sqrt{(t^2 \cdot \cos(2 \cdot \alpha) + 2 \cdot b^2 - t^2)} - \sqrt{2} \cdot t \cdot (2 \cdot a^2 + b^2 - t^2) \cdot \cos(\alpha))}{2 \cdot (b^2 - t^2)}$$

Substracting the right sides of #15 and #18 results in 0 verifying the identity.

#15 (and #18 of course) is only valid for $t \neq b$ – which is a circle passing the origin – The shape of the image curve depends on the position of the pre-image circle. It has a vertex if it is osculating the pole circle. Then $t = a\sqrt{2} + b$ or $t = a\sqrt{2} - b$. In the first case we will find "tears" or "drops" left of $x = 0$, in the other case "tears" right of $x = 0$ ($2b < a\sqrt{2}$) or cardioids ($2b > a\sqrt{2}$) will appear. See the following plots. The radius of the inversion circle $a = 2$, the pole circle is the dashed line – the plot of #4 from above, the pre-image circle is in red and its image in blue.

For ploting the dashed circle one has to switch – for only this single plot to rectangular coordinates and set the parameter interval to [0,1]. All plots are done applying the slider bar feature of DERIVE 6.
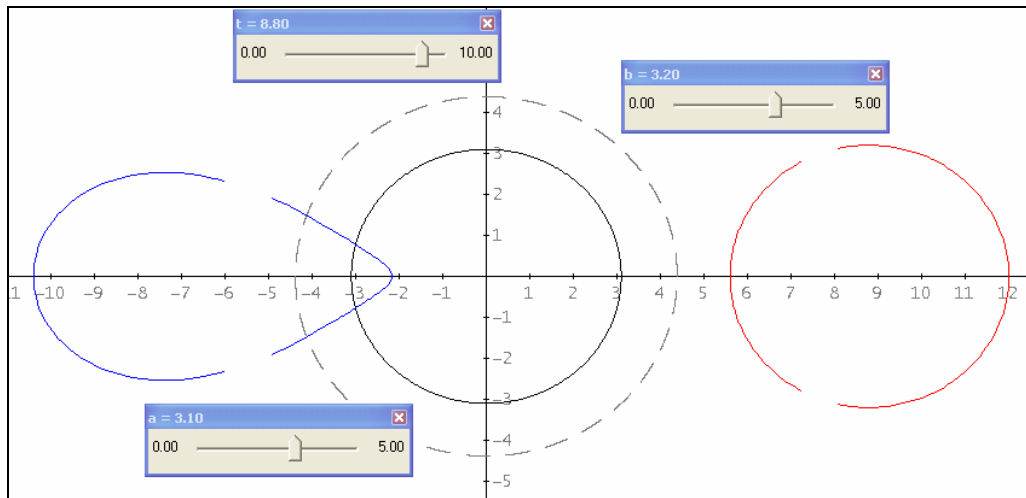

This is the plot with the sliders in any position.

The plot performed with the DOS-versions didn´t show any gaps. (I tried this with versions 2.xx and 3.xx.) For circles which don´t contain the origin within their circumference the recent Derive version shows gaps in the region of the osculating points between the circle and its tangents from the origin. The gaps are appearing in the mapping, too.

(We will compare the representations given by other CASs at the end of this contribution.)
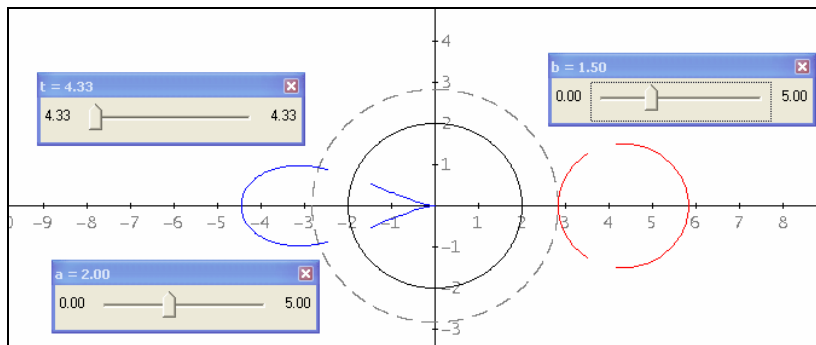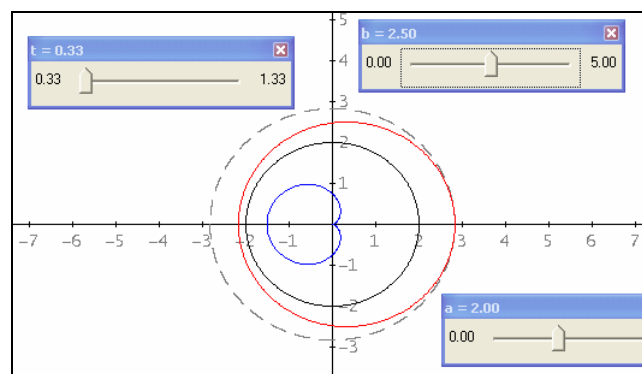

$$t = 2\sqrt{2} + 1.5; b = 1.5$$
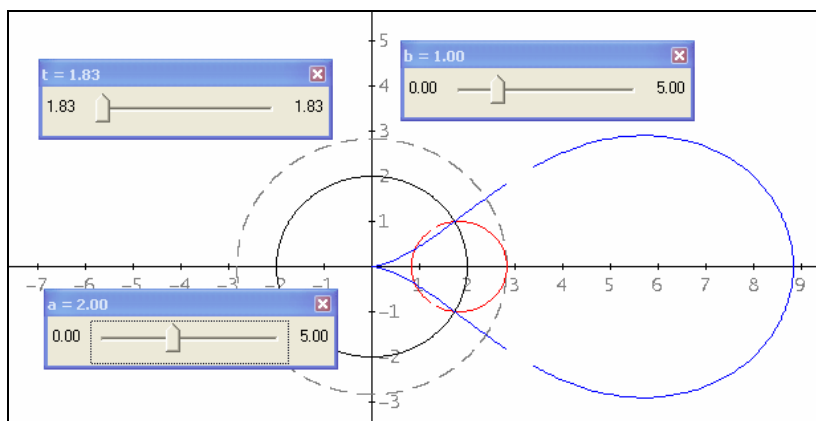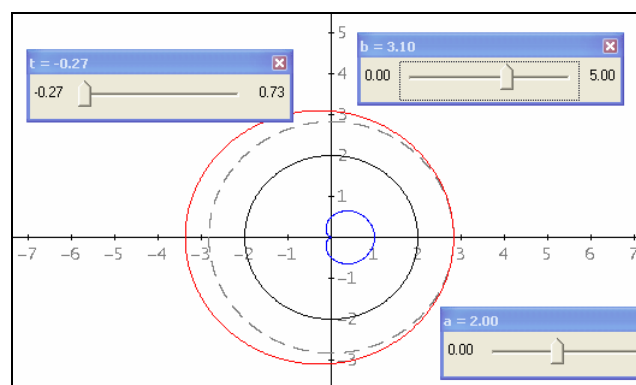

$$t = 2\sqrt{2} - 1; b = 1$$

Later we will see that for $2b = a\sqrt{2}$ the "tear" will degernate to a kissoid. For increasing b the kissoid will change to a cardiod. If b keeps increasing then the cardioid will shrink and for $b = a\sqrt{2}$ degenerate to the origin. $b > a\sqrt{2}$ will result in a cardiod right of $x = 0$.
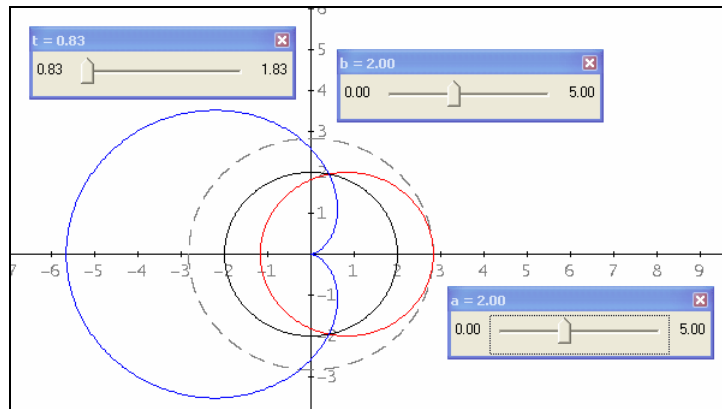
$$t = 2\sqrt{2} - 2; b = 2$$

$$t = 2\sqrt{2} - 2.5; b = 2.5$$

$$t = 2\sqrt{2} - 3.1; b = 3.1$$

If the pre-image circles intersects the pole circle then the image curve has a double point in the origin. It has the shape of a leminscate or of a snail of Pascal. Because symmetric lemniscates are intersecting the x-axis symmetrically wrt the origin we substitute $\alpha = 0$ and then $\alpha = \pi$ in expression #15 above in order to obtain #18 and #19:

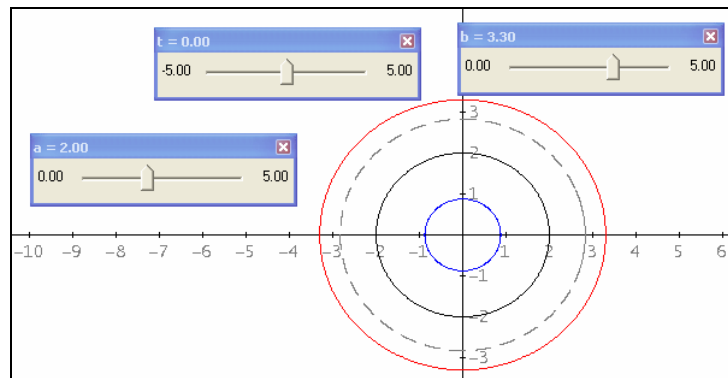#18: $r = \dfrac{(2 \cdot a^2 - b^2 + t^2) \cdot |b| - t \cdot (2 \cdot a^2 + b^2 - t^2)}{b^2 - t^2}$

#19: $r = \dfrac{(2 \cdot a^2 - b^2 + t^2) \cdot |b| + t \cdot (2 \cdot a^2 + b^2 - t^2)}{b^2 - t^2}$

#20: $\text{SOLVE}\left(\dfrac{(2 \cdot a^2 - b^2 + t^2) \cdot |b| - t \cdot (2 \cdot a^2 + b^2 - t^2)}{b^2 - t^2} = \dfrac{(2 \cdot a^2 - b^2 + t^2) \cdot |b| + t \cdot (2 \cdot a^2 + b^2 - t^2)}{b^2 - t^2}, t \right)$

#21: $t = -\sqrt{(2 \cdot a^2 + b^2)} \; \lor \; t = \sqrt{(2 \cdot a^2 + b^2)} \; \lor \; t = 0$

The two values for *r* should be equal because of the given symmetry. Then we solve the equation for the possible distances of the centre of the pre-image circle.

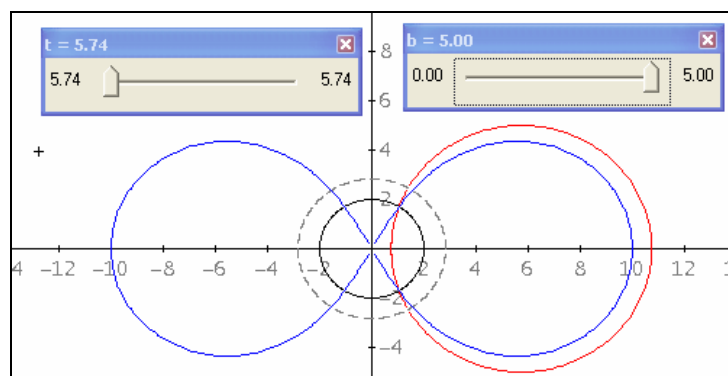*t* = 0 gives a circle concetric to the inversion circle.
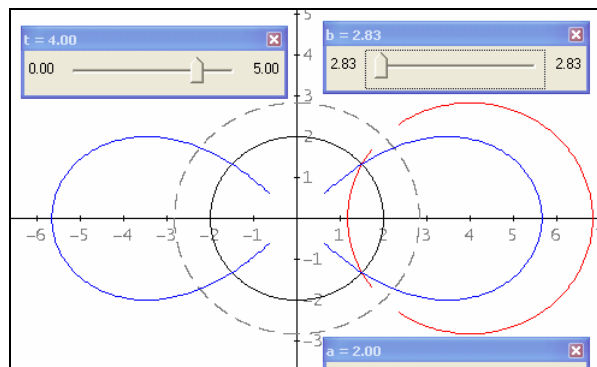

$$a = 2; b = 3.30; t = 0$$

Substituting $t = \sqrt{2a^2 + b^2}$ in #15 and simplifying gives the equation of a lemniscate (#23) and if the tangents in the double point are perpendicular ($b = a\sqrt{2}$) then it will become the Lemniscate of Bernoulli. If *b* decreases then the lemniscate will become smaller.

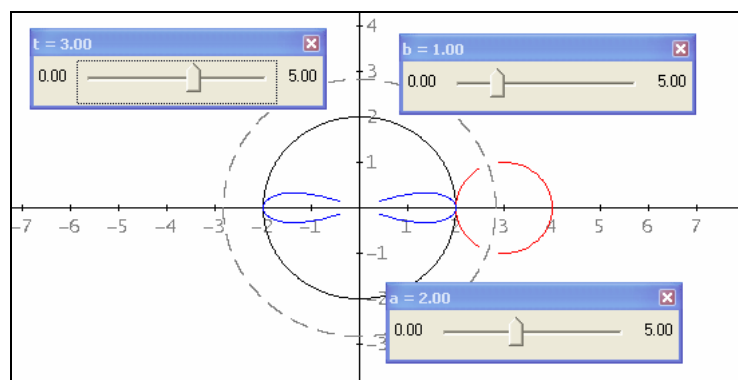#23: $r = -2 \cdot \sqrt{((2 \cdot a^2 + b^2) \cdot \cos(\alpha)^2 - 2 \cdot a^2)}$

#24: $r = -2 \cdot \sqrt{2} \cdot |a| \cdot \sqrt{(2 \cdot \cos(\alpha)^2 - 1)}$
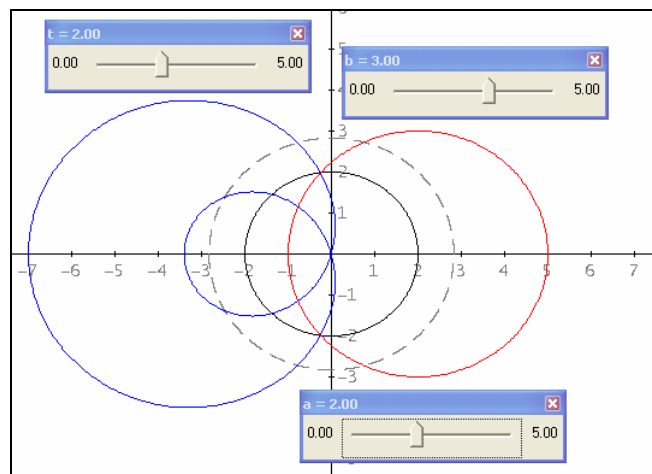

$$a = 2; b = 5; t = \sqrt{33}$$

$$a = 2; b = 2\sqrt{2}; t = 4$$

Lemniscate of Bernoulli

$$a = 2; b = 1; t = 3$$

A kind of snail of Pascal appears if the pre-image circle intersects the pole circle and contains the origin. The general form of this curve is $r = a + b \cos \alpha$. It is a pity that it is not possible to obtain this curve by finding a special form of #18.

$$a = 2; b = 3; t = 2$$

3.   **If the pre-image circle passes the origin then $t = b$ and the image curve contains infinity.**

For $t = b$ the equation of this special family of circles is given by #27:

$$\text{\#25:} \quad ss := t \cdot \cos(\alpha) + \sqrt{(b^2 - t^2 \cdot \sin(\alpha)^2)}$$

$$\text{\#26:} \quad ss := |b \cdot \cos(\alpha)| + b \cdot \cos(\alpha)$$

#27: $ss := 2 \cdot b \cdot COS(\alpha)$

#28: $r = \dfrac{2 \cdot a^2}{ss} - ss$

#29: $r = \dfrac{a^2}{b \cdot COS(\alpha)} - 2 \cdot b \cdot COS(\alpha)$

Simplifying the equation of the mapping gives #29 which is the image of these special circles. It is remarkable that this equation is identical with expression #5 from page 35 – the image curves of straight lines – if $b \cdot t = - a^2$. That means: to each line exists one circle passing the origin with the same image curve. Geometrically you can obtain the centre of the circle by finding the pole of the line wrt the inversion circle and then reflecting it wrt the origin. The kissoid, the strophoid and the trisectrix can be produced as pictures of circles.

Substituting $b = \dfrac{a}{\sqrt{2}}$, $b = a$ and $b = a\sqrt{2}$ in #29 leads to the said curves.
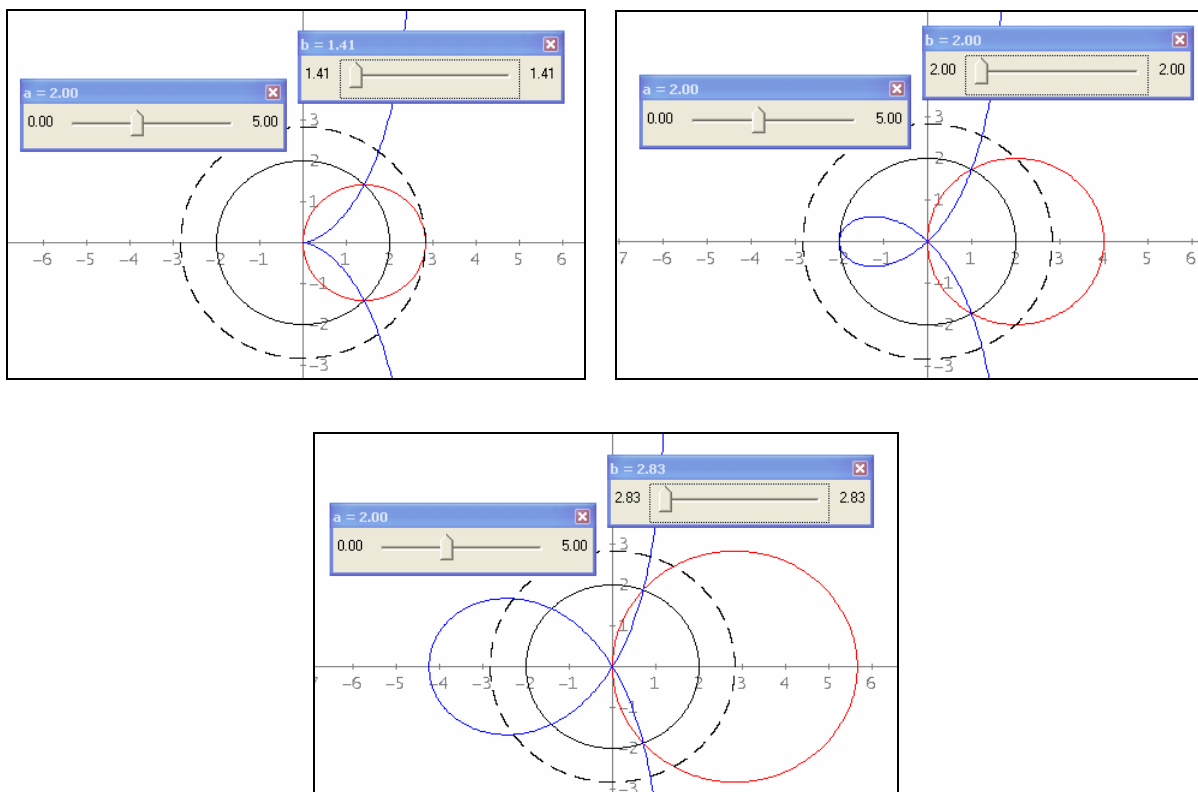

It is fascinating to watch the various changes from one type of curve to another by appropriate variation of the parameters. You can hardly find limits for playfulness with *DERIVE*. (*The deficit of the weak representation of the polar forms is more than substituted by the advantage of introducing slider bars, Josef in 2009*)

A final remark for teachers: surely plotting equations is a comfortable application of the computer. But primarily we have to train the students´ ability to imagine the graph of a given equation in the various systems of coordinates and to recognize its dependency on the parameters of the equation. For this purpose you have to practise sketching the curves by hand, too. After having acquired some experience the use of the computer is a meaningful enrichment and completion. Peter Baum.

Preparing this contribution for the DNL I reproduced Peter Baum´s plots and I had the idea to represent the *pole circle* as a dashed line (for $0 \le \Phi \le 1$ and in rectangular coordinates) and the pre-image circle as a bold line. Here are the two *DERIVE*-tools with three examples followed by the *DERIVE* screen. Josef

#1: $\text{KDOP}(m, n, r, sc := 1) := \begin{bmatrix} m + (r - 0.02 \cdot sc) \cdot \cos(2 \cdot \pi \cdot \phi) & n + (r - 0.02 \cdot sc) \cdot \sin(2 \cdot \pi \cdot \phi) \\ m + (r + 0.02 \cdot sc) \cdot \cos(2 \cdot \pi \cdot \phi) & n + (r + 0.02 \cdot sc) \cdot \sin(2 \cdot \pi \cdot \phi) \end{bmatrix}$

#2: $\text{KSTRX}(m, n, r, t, p) := \text{VECTOR}\left(\left[m + r \cdot \cos\left(\frac{2 \cdot \pi}{t} \cdot (i + p \cdot \phi)\right), \ n + r \cdot \sin\left(\frac{2 \cdot \pi}{t} \cdot (i + p \cdot \phi)\right)\right], \ i, \ 0, \ t - 1\right)$

#3: $\text{KDOP}(2, 1, 1.5)$

#4: $\text{KSTRX}(-1, -1, 1.8, 40, 0.8)$

#5: $\text{KSTRX}(2.5, -1.5, 1.2, 20, 0.5)$

#6: $\text{KSTRX}(2.5, -1.5, 1.23, 20, 0.5)$
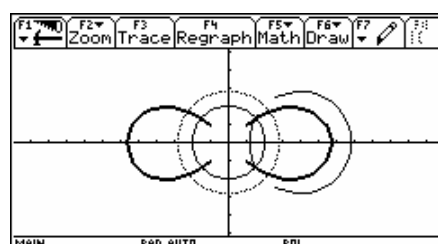
#7: $\text{KSTRX}(-2, 1, 1.5, 4, 0.75)$
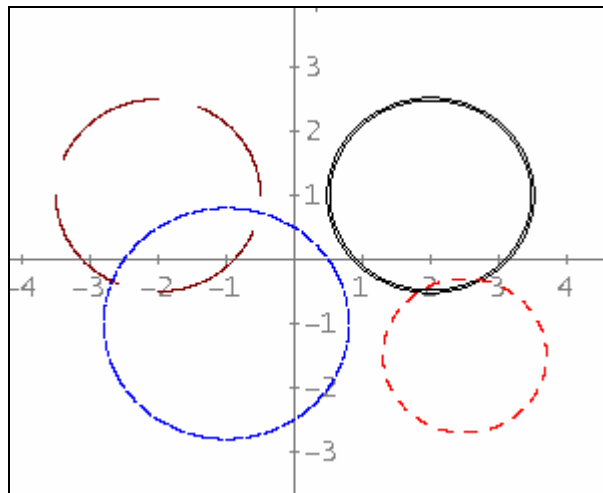
Let´s look how other Computer Algebra Systems of 2009 are performing the plots of the curves in polar representation:

The Voyage 200 behaves pretty the same like DERIVE, we see the gaps.

TI-NspireCAS has no problems – like the V200 and the TI-92 to do the calculations but the plots are a bit worse because the endpoints of the gaps are connected – which has a mathematical explication but does not look fine. I introduced slider bars which are not available on the TI-92 and Voyage 200:

$$r := \frac{2 \cdot a^2}{s} - s$$

$$s := t \cdot \cos(\theta) + \sqrt{b^2 - t^2 \cdot (\sin(\theta))^2}$$

$$r \qquad \frac{-\left(2 \cdot \cos(\theta) \cdot t \cdot \sqrt{b^2 - (\sin(\theta))^2 \cdot t^2} + \left(2 \cdot (\cos(\theta))^2 - 1\right) \cdot t^2 - 2 \cdot a^2 + b^2\right)}{\sqrt{b^2 - (\sin(\theta))^2 \cdot t^2} + \cos(\theta) \cdot t}$$

I chose the minimized form of the slider bars.
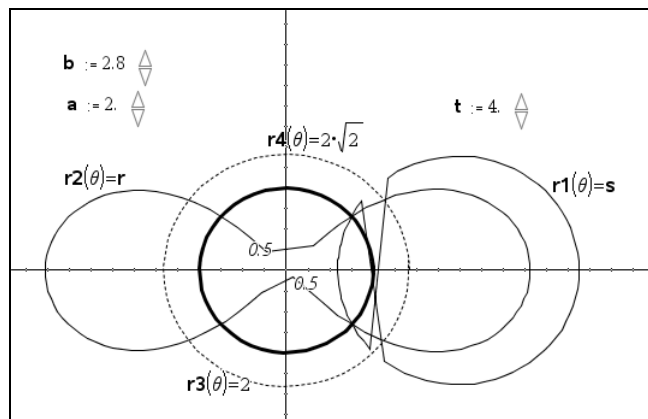
That it can be done better shows the plot produced with WIRIS (including slider bars, too):

$$r = \frac{2a^2}{s} - s;$$

$$s = t \cdot \cos(\alpha) + \sqrt{b^2 - t^2 \cdot \sin(\alpha)^2};$$

$$r \rightarrow \frac{-2 \cdot t \cdot \cos(\alpha) \cdot \sqrt{-t^2 \cdot \sin(\alpha)^2 + b^2} + (-t^2 \cdot \cos(\alpha)^2 + t^2 \cdot \sin(\alpha)^2) + (2 \cdot a^2 - b^2)}{\sqrt{-t^2 \cdot \sin(\alpha)^2 + b^2} + t \cdot \cos(\alpha)}$$

```
a:=slider(0..5..0.2);b:=slider(0..5..0.1);t:=slider(-10..10..0.5);
plot({a,b,t}) → plotter1
plot(polar_curve(a,α,0..2π)) → plotter1
plot(polar_curve(a·√2,α,0..2π)) → plotter1
plot(polar_curve(s,α,0..2π),{color=red}) → plotter1
plot(polar_curve(r,α,0..2π),{color=blue,line_width=3}) → plotter1
```
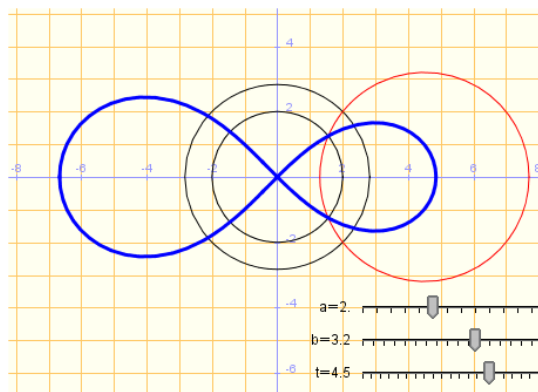
Raymond Wieloszynski, Six-Fours les Plages, France

Monsieur,

I desire to pose a question, please excuse for my English:

How to develop by series entieres in zero the next equation differentielle:

4x (1 – x) y" + 2(1 – 3x) y' – y = 0

or this equation differentielle in serie entiere solution of this equation y(0) = 1

x y" + (1 – 2x) y' – 2y = 0

I thank you for your reply.

**DNL:** I am not an expert in this field but I looked into the manual of DERIVE 3.xx and found a function in the utility file ODE_APPR.MTH which might help you (even with DERIVE 2.50).
I add a listing for the generalized problem and copy the respective explanation of the function from the DERIVE manual (now from the DERIVE 6.10 online help!):

TAYLOR_ODE2 (r, x, y, v, x0, y0, v0, n) simplifies to an nth order truncated Taylor series solution of the equation y" = r(x, y, v). Here v represents y', and v0 is the value of v at x = x0. For example, TAYLOR_ODE2 (2·x·v + x²·y + 3·x, x, y, v, 0, 0, 1, 5) simplifies to

$$\frac{3 \cdot x^5}{10} + \frac{5 \cdot x^3}{6} + x$$

To verify that this is an approximate series solution to the differential equation, substitute this series solution for y and substitute the derivative of this expression for v into the unsimplified difference between the two sides of the equation. Then use the Calculus Taylor Series command to verify that the third order Taylor series approximation of that difference is 0. The reason for using a third rather than fifth order Taylor series approximation is that the second derivative of the truncated series reduces its order by 2.

TAYLOR_ODE2 simplifies to ? if r is not sufficiently differentiable for an nth order truncated series. Try a smaller value for n if this happens.

Now let´s have a try:

#1: $\text{TAYLOR\_ODE2}\left(\dfrac{y - 2 \cdot (1 - 3 \cdot x) \cdot v}{4 \cdot x \cdot (1 - x)}, \ x, \ y, \ v, \ 0, \ y0, \ v0, \ 3\right)$

#2:   ?

#3: $\text{TAYLOR\_ODE2}\left(\dfrac{y - 2 \cdot (1 - 3 \cdot x) \cdot v}{4 \cdot x \cdot (1 - x)}, \ x, \ y, \ v, \ x0, \ y0, \ v0, \ 3\right)$

#4: $$\frac{x^3 \cdot (2 \cdot v0 \cdot (14 \cdot x0^2 - 9 \cdot x0 + 3) + y0 \cdot (7 \cdot x0 - 3)) - 3 \cdot x0 \cdot x^2 \cdot (2 \cdot v0 \cdot (20 \cdot x0^2 - 17 \cdot x0 + 5) + y0 \cdot (9 \cdot x0 - 5)) + \sim}{\sim}$$

$$\frac{3 \cdot x0^2 \cdot x \cdot (2 \cdot v0 \cdot (34 \cdot x0^2 - 41 \cdot x0 + 15) + y0 \cdot (11 \cdot x0 - 7)) - x0^2 \cdot (2 \cdot v0 \cdot x0 \cdot (56 \cdot x0^2 - 81 \cdot x0 + 33) - y0 \cdot (35 \sim}{48 \cdot x0^2 \cdot (x0 - 1)^2}$$

$$\cdot x0^2 - 87 \cdot x0 + 48))$$

The first question mark is the answer to use $x = 0$ but for $x = 0$ function r is not defined. So I did once more and generalized: $y(x0) = y0$ and $y'(x0) = v0$.

Just to demonstrate how the function works I try with $y(x = 2) = 0$, $y'(x = 2) = 1$. I ask for the series of order 5:

#5: $\text{TAYLOR\_ODE2}\left(\dfrac{y - 2 \cdot (1 - 3 \cdot x) \cdot v}{4 \cdot x \cdot (1 - x)}, \ x, \ y, \ v, \ 2, \ 0, \ 1, \ 5 \right)$

#6: $\dfrac{1535 \cdot x^5 - 17298 \cdot x^4 + 79608 \cdot x^3 - 189136 \cdot x^2 + 238128 \cdot x - 128928}{6144}$

This was all what I could do in 1995. In the meanwhile **Terence Etchells** provided a function TaylorODE which can be found in the Users\EquationSolving subdirectory of Derive:

TaylorODE.dfw   Defines the function TaylorODE(f, o, t, h, n, init_vals, header?, xy_only?) that solves initial value Ordinary Differential Equations of any order using Taylor's method.

#15: $\left(\text{TaylorODE}\left(\dfrac{y0 - 2 \cdot (1 - 3 \cdot x) \cdot y1}{4 \cdot x \cdot (1 - x)}, \ x, \ y, \ 2, \ 5, \ 0.1, \ 10, \ [2, \ 0, \ 1] \right) \right)\downarrow\downarrow[1, \ 2]$

#16: $\left(\text{TaylorODE}\left(\dfrac{y0 - 2 \cdot (1 - 3 \cdot x) \cdot y1}{4 \cdot x \cdot (1 - x)}, \ x, \ y, \ 2, \ 5, \ -0.1, \ 10, \ [2, \ 0, \ 1] \right) \right)\downarrow\downarrow[1, \ 2]$
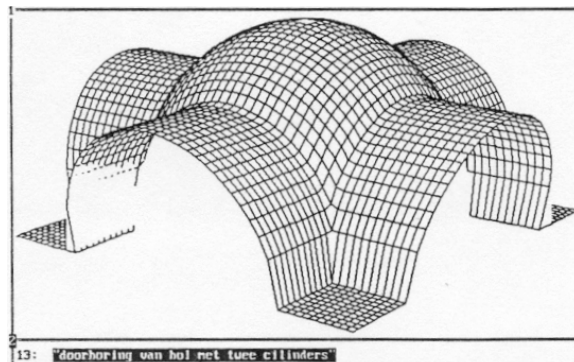
I plot function #6 and the points created by #15 and #16 on the same axes:
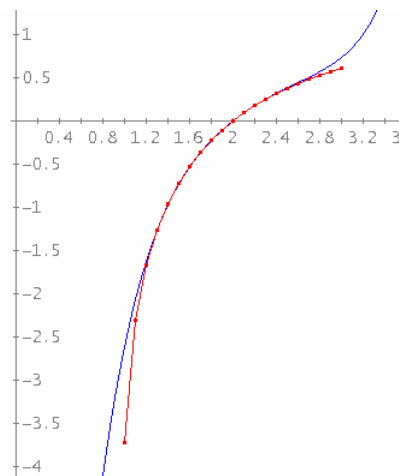
(The respective file is among the DNL#19-files.)
Maybe that any of our members will find a more suitable answer?

13: "doorhoring van bol met twee cilinders"

ISAAC 1997

From: rcorless@rawhide.cecm.sfu.ca (Rob Corless)
Subject: For distribution to the DERIVE User Group

We (Bruce Char and Rob Corless) are considering putting in a bid in to hold ISSAC 97 in Hawaii. There is little doubt in our minds that most potential attendees would find Hawaii a pleasant venue in which to hold a meeting. Our preliminary investigations into local arrangements and travel makes us believe it will be a satisfactory venue from those points of view as well.

Our current thinking has centered on Kihei Maui, on the shore of southeastern Maui, a few miles from the Maui Supercomputing Center. We have a quote from the Maui Intercontinental Hotel of approximately $100 US per night per room, single or double. There are accommodation alternatives (e.g. rental of condominiums that house four or six) in the area of Kihei as well. Living expenses are otherwise comparable to the continental USA. There are regularly scheduled commercial flights direct from the US mainland to Maui; ground transfer from the Maui airport to Kihei would be approximately 30 minutes via van.

Airfare information at this early date is sketchy and vague; the current "21 day advance reservation" price is approximately $1100 US return from Philadelphia to Maui; from Vancouver, with a block booking of 100 seats, the price is $499 (Canadian) return to Honolulu. Return airfare between Honolulu and Maui is approximately $150 US. Qantas flies from Sydney to Honolulu for $1295 (Australian). The fact that Canadian prices appear to be the cheapest above may reflect the fact that a Canadian travel agent did most of the price search; cheaper flights may be available through your local agent. (For example last year during a sale, the return airfare between Philadelphia and Honolulu was approximately $700). The agent was unable to give quotes for flights from Tokyo or from Europe, which would clearly be useful to know.

Expressions of support have been received for our bid both by the Maui Supercomputing Center and the Soft Warehouse (creators of Derive). They would assist us in making local arrangements. The Supercomputing Center has extensive prior experience in setting up computing meetings at the Intercontinental Hotel.

There is also the possibility of running another meeting (Stanly Steinberg's Applied Computer Algebra meeting) 'back to back' with ISSAC, to optimize the use of air transportation and clout with hotel bookings. We would appreciate suggestions for other meetings that we try to could co-schedule with ISSAC '97.

We are also strongly interested in encouraging people who think of themselves as numerical analysts to come to this meeting. Some ISSAC meetings in the past have had a numerical flavour; with the Maui High Performance Computer Center supporting the meeting, it seems quite reasonable to expect that this meeting will have a similar sub-theme of numerical/symbolic interaction.

Please take the time to respond to the following questionnaire.

1. If ISSAC '97 was held on one of the Hawaiian islands, would you be interested in attending?
2. Would you be able to come?
3. Block-bookings or charter flights may significantly lower the cost of airfare. This requires planning ahead Would you be able to do so?
4. The usual date for ISSAC is in the (northern hemisphere) summer. The climate of Hawaii is uniformly pleasant. and a summer conference there would provide a welcome change for those living in hot, muggy areas in (for example) the U.S. and Canada. However, several people have indicated informally that they would like to have an ISSAC meeting in Hawaii during the (northern hemisphere) winter, say January or February. How do you feel about that?
5. Would you classify yourself as a numerical analyst or a computer algebraist (for want of a better name for people interested in symbolic computation) or both or neither?
6. Please give us any other comments you have.

Send all responses to bchar@mcs.drexel.edu or rcorless@uwo.ca

Thank you for your attention

Bruce Car and Rob Corless

Dear Josef

A student recently submitted an essay on the use of DERIVE in mathematics education. I have enclosed an amusing quote from his essay, that may appeal to readers of the DERIVE Newsletter.

"Tweedledum and Tweedledee agreed to have a battle,

For Tweedledee said Tweedledum had spoiled his nice new rattle,

Just then flew by a monstrous crow

As big as a tar barrel

Which frightened both our heroes so

They quite forgot their quarrel."

And there in nutshell we have DERIVE and its acolytes depicted as a monstrous crow while the arguments for and against such primitive toys as calculators are cut off by the arrival of this terrifying creature.

Best wishes
Ted Graham